Universidade de Brasília – UnB
Faculdade de Tecnologia – FT
Departamento de Engenharia Elétrica – ENE
Laboratório de Redes de Comunicação - LabRedes

Projeto Final de Graduação

Definição e Implantação de Ambiente de Colaboração Visual com Ferramentas de Software de Código Aberto

Aluno: Daniel Cardoso Danna
Matrícula: 96/04898
Orientadores: Prof. Ricardo Staciarini Puttini
Prof. Rafael Timóteo de Sousa Júnior

Julho 2001

Agradecimentos

Aos professores e orientadores deste estudo, Rafael e Ricardo, pela seriedade e dedicação com que conduziram o trabalho, bem como o grande apoio prestado nos momentos mais delicados das implementações.

Ao pessoal do LabRedes pela ajuda prestada na instalação dos software, pelo auxílio na realização dos testes, e principalmente pela amizade.

Aos que me apoiam ou já me apoiaram em outras fases da minha vida, pois sem dúvida, me ajudaram a enxergar o amor para com o próximo e me fizeram viver momentos inesquecíveis.

Aos meus eternos amigos, sempre presentes nas horas mais difíceis e que provam a cada dia que passa, o valor incalculável de um ombro companheiro.

Às minhas irmãs, pela união que mostram nos momentos felizes e tristes, por saber que posso contar com vocês em qualquer situação, e por mostrarem que a família é o único bem que não passa na vida. Amo vocês.

À minha mãe querida, pelo exemplo de mulher, dedicação, sinceridade, afeto, companheirismo, honestidade, força, justiça e amor. Seus exemplos ficarão guardados para sempre comigo e sua lembrança será eterna. Sua preocupação excessiva me fez entender o significado do que é ser mãe e do que é amar um filho. Com certeza, sem todas estas lições que me foram passadas, eu não conseguiria alcançar a glória da graduação neste momento. Você me faz entender o sentido da vida. Te amo para sempre, mesmo sem a sua presença.

Ao meu pai querido, pelo exemplo de profissionalismo, superação, honestidade, sentimento, competência, afeto, orientação e amor. Você sempre foi e será o meu ídolo, suas atitudes só me fazem te admirar cada vez mais. Obrigado por todo carinho e amor que você sempre me dá, além do conforto de ter você sempre ao meu lado. Espero um dia ser tão bom e grandioso como você. Pai é aquele que apoia o filho independente da situação, que comemora com cada vitória alcançada e que tem uma palavra amiga quando é necessário. Você é o meu maior tesouro e espero um dia retribuir em dobro tudo que você fez por mim. Te amo pai.

Ao Nosso Senhor, o meu pastor, que nada em minha vida deixará faltar. Sou seu servo e por isso faça de mim instrumento de vossa paz. Somente tu Senhor, para me fazer entender que as perdas são para o bem e crescimento, não para o mal. Acredito e sempre acreditarei na tua bondade.

Resumo

Este trabalho apresenta a definição e montagem de um ambiente de colaboração visual para *Desktop* baseado em software de código aberto. Este tipo software é particularmente conveniente para implementações acadêmicas, uma vez que é distribuído gratuitamente, e pode ainda ter seu código alterado de acordo com a necessidade de quem o utiliza. Assim, o ambiente montado deverá servir para o desenvolvimento de tecnologia associada aos padrões de videoconferência existentes.

O projeto foi desenvolvido baseado na implementação da Recomendação ITU-T H.323, realizada no contexto do projeto OpenH323 que define uma série de entidades, seus modos de funcionamento, interoperabilidade entre eles, além de aspectos como conversão de áudio, vídeo e dados. No contexto deste trabalho são implementados e testados terminais, *gatekeepers* e controladores multiponto (MCU). Também é apresentado um analisador de protocolos para os planos de dados e de controle da pilha H.323.

Abstract

This work presents a definition and construct of a visual collaboration environment for Desktop based in opensource software. This is particularly convenient for academics implementation, the software is freely distributed, and its code can be modified according to the needs of the user. So, the developed environment will improve technology associated to the existing videoconferencing standards.

The project was developed based on H.323 ITU-T Recommendation, realized in the context of OpenH323 project, which defines a series of entities, its operating modes, interoperability and aspects like audio, video and data conversion. In the context of this work, terminals, gatekeepers and Multipoint Control Unit (MCU) are implemented and tested. Furthermore, a protocol analyzer for data plane and H.323 stack control is presented.

Índice

Lista de Tabelas

Lista de Figuras

Lista de Acrônimos

BAS	Bit rate Allocation Signal
CCA	Chair Command Acquire
CCD	Chair Command Disconnect
CCK	Chair Command Kill
CIC	Chair-control Indicate Capability
CIS	Chair Indicate Stopped-using
CODEC	Coder-Decoder
FEC	Forward Error Correction
IIS	Information Indicate String
ISDN	Integrated Services Digital Network
ITU	International Telecommunication Union
ITU-T ITU	Telecommunication Sector (formerly CCITT)
MCC	Multipoint Command Conference
MCN	Multipoint Command Negating MCS
MCS	Multipoint Command Symmetrical Data-transmission
MCU	Multipoint Control Unit
MCV	Multipoint Command Visualization-forcing
MIS	Multipoint Indicate Secondary-status
MIV	Multipoint Indicate Visualization
MIZ	Multipoint Indicate Zero-communication
MLP	Multilevel Protocol Channel
PCM	Pulse Code Modulation
RAN	Random Number
SBE	Single Byte Extension
SCM	Selected Communication Mode
TCI	Terminal Command Identify
TCP	Terminal Command Personal Identifier
TCU	Terminal Command Update
TIA	Terminal Indicate Assignment
TIC	Terminal Indicate Capability

TID	Terminal Indication Dropped
TIF	Terminal Indicate Floor-request
TII	Terminal Indicate Identity
TIL	Terminal Indicate List
TIN	Terminal Indicate Number
TIP	Terminal Indicate Personal Identifier
TIS	Terminal Indicate Secondary
TIX	Terminal Indicate Additional Channel X
TCS	Terminal Command String
VCB	Video Command Broadcast
VCF	Video Command "Freeze-picture request"
VCS	Video Command Select
VIN	Video Indicate Number

1. Introdução

A Internet mundial vem se afirmando cada vez mais como um dos principais meios de comunicação e distribuição de informação disponíveis na atualidade. O modelo de conectividade simples, mas robusto, e a disponibilização da tecnologia e de ferramentas que o implementam, a um baixo custo e em larga escala, foram, sem dúvida, alguns dos principais fatores que contribuíram para disseminação e popularização acentuada dessa rede. No entanto, um novo ciclo de desenvolvimento e de emprego de tecnologia da informação relacionado com a Internet vem se formando, em especial a partir do surgimento de redes de alta velocidade. Este cenário de interconexão em altas taxas vem permitindo a concepção e a implementação de novas aplicações de rede, em especial com a utilização cada vez mais significativa de recursos e ferramentas de multimídia. Essa nova fase da Internet, que inclui o desenvolvimento e a implantação de redes de alta velocidade e a concepção e disseminação de novas aplicações, consiste o que se chama hoje de Internet2 [3].

No contexto das novas aplicações de Internet, um dos principais focos de desenvolvimento e aporte tecnológico está relacionado com a imersão de recursos de multimídia interativa. Assim, uma grande diversidade de aplicações tem sido apoiada e até mesmo fundamentada pela utilização de recursos de comunicação e interação entre indivíduos ou entre um grupo de indivíduos, fisicamente posicionados em locais diferentes, envolvendo mídias de áudio, vídeo e dados em geral, constituindo um ambiente de videoconferência ou mesmo de colaboração visual [1].

As soluções de videoconferência vem sendo bastante estudadas e desenvolvidas, podendo ser classificadas em dois tipos básicos: a solução *Desktop* e a solução de estúdio.

Solução *Desktop* é aquela na qual o indivíduo utiliza um computador pessoal com acesso à rede (modem ou placa de rede local), e realiza a videoconferência com recursos de software e hardware *off shelf* e de baixo custo (como o Netmeeting da Microsoft® ou OpenH.323 utilizado neste trabalho juntamente com *webcams*, placa de vídeo, placa de som, microfone), tendo o próprio computador como unidade de processamento e tratamento da informação e de controle da comunicação. Este ambiente pode ser construído com relativa facilidade e os custos envolvidos não são altos. Como, em geral, os computadores são comercializados com todos estes periféricos, os usuários têm acesso aos recursos necessários para realização de videoconferência Desktop [6].

Já a solução em estúdios especialmente preparados, é aquela na qual a videoconferência é realizada a partir de um local com ambientação (isolação acústica, iluminação, cenário, mobiliário, refrigeração) e equipamentos (câmeras de alta definição, monitores de TV, recursos avançados de apresentação, etc.) especialmente projetados para este fim, podendo resultar em um ambiente de videoconferência de alta qualidade. No entanto, o investimento em infra-estrutura e equipamentos (*hardware* e *sofware*), é bem mais elevado que no caso anterior. Assim, os custos para montagem desse tipo de ambiente podem ser impeditivos para um número elevado de usuários/aplicações [6].

Do ponto de vista formal, essas tecnologias estão padronizadas em um conjunto de recomendações do ITU-T, denominada série H. Em especial, interessa no contexto deste trabalho o conjunto de recomendações H.323 relativo a serviço de comunicação multimídia em redes de pacotes que não provêem qualidade de serviço. A Internet é um exemplo típico deste tipo de rede, e seu amplo uso em nível mundial, constitui um bom motivo para que a recomendação H.323 seja ainda mais pesquisada e aprimorada [5].

Várias ferramentas que implementam a pilha H.323 vêm sendo desenvolvidas, entre elas soluções de *hardware* e *software.* Basicamente, soluções de *hardware* são bastante utilizadas em ambientes de estúdio e são manipuladas controlando-se parâmetros estabelecidos pelo seu fabricante. Apresentam alta capacidade de processamento, uma vez que possuem circuitos dedicados e especialmente projetados para este fim. Já as soluções *Desktop* utilizam basicamente *software*; estes possuem maior escalabilidade, sua atualização demanda menos trabalho, e apresentam preço variável, devido a existência de soluções pagas ou mesmo *freeware*.

Este trabalho visa a definição e implementação de ambiente de colaboração visual H.323 compatível, a partir de soluções de software de código aberto, em um modelo *Desktop*. Esta escolha esta relacionada com resultado de esforço para definição de um ambiente de desenvolvimento de tecnologia H.323. As soluções tecnológicas que utilizam *software* livre são particularmente interessantes em ambientes acadêmicos, pois permitem a definição de soluções de baixo custo, podem ser distribuídos e utilizados gratuitamente, bem como com acesso transparente ao conhecimento envolvido nas implementações, uma vez que o código fonte está disponível. Isto permite ao usuário incluir novas funções ao sistema ou mesmo retirar algumas, visando ganho de performance da rede.

Assim, desenvolveu-se um sistema de videoconferência, a partir de aplicativos de *software* aberto e utilizando arquiteturas de redes IP, que é o protocolo de comunicação da Internet. Buscou-se um ambiente no qual os seus componentes puderam todos serem implementados por meio de

software, sob a plataforma Linux, que é um sistema operacional de código aberto. Objetivou-se, também, a comunicação não só entre dois terminais, mas a comunicação entre diversos terminais simultaneamente.

O ambiente constitui-se de estações clientes (terminais H.323 [35]) dotadas de equipamento de captura de vídeo do tipo *webcam* e interfaces de áudio padrão, que realizam a comunicação em videoconferência em um backbone de Intranet/Internet simulado, interligadas por um *firewall*, que realiza o controle de acesso e de tráfego entre elas. São utilizados ainda *gatekeepers* [35], que provêem conversão de endereços e controle de acesso e de banda para os clientes, uma unidade de controle de videoconferência multiponto (MCU) [35] e um servidor de *streaming* [9], descritos com maiores detalhes a seguir. Está prevista também a implantação de *gateways* com as redes públicas de telefonia (*gateway* H.323/H.324) e a rede digital de serviços integrados – RDSI (gateway H.323/H.320) [35], não realizada no escopo deste trabalho. Vale ressaltar que todos os componetes citados estão implementados em software e utilizam hardware *off shelf* (computadores tipo IBM PC com *webcams*, placas de som, placas de captura de vídeo, placas de rede, modems).

As implementações de software usadas no ambiente são derivadas de uma biblioteca de software aberto denominada OpenH323, disponível através de uma licença pública MPL (Mozilla Public License)[8]. Esse tipo de licença é caracterizada pela disponibilidade do código fonte do software, aliada a concessão de direito de modificações ou adaptações sem necessidade de autorização prévia do produtores originais do software, o que caracteriza esse tipo de implementação como software livre [12]. A MPL normaliza ainda a utilização em produtos comerciais e revendas. Mais especificamente, OpenH323 é um projeto desenvolvido e coordenado por uma empresa australiana, a Equivalence Pty Ltd, aberto a qualquer pessoa interessada. Os códigos fonte dos produtos e bibliotecas desenvolvidos estão acessíveis na *homepage* do projeto[10], e atualizações nos arquivos são constantemente publicadas a partir de um sistema público de controle de versão (CVS) [10].

No capítulo 2 é apresentado uma descrição da Recomendação H.323, seus benefícios, o estabelecimento de chamadas, além de protocolos de funcionamento. O capítulo 3 apresenta as entidades envolvidas em uma chamada H.323, seguindo para o capítulo 4 que apresenta o servidor de *streaming* e suas funcionalidades. Finalmente, no capítulo 5 é proposto um backbone experimental para comunicação a distância e no capítulo 6 são apresentados os testes realizados com as entidades da rede, propostas para futuras implementações inclusive com qualidade de serviço garantida. As conclusões acerca do trabalho são apresentadas no capítulo 7.

2. A Recomendação H.323

A recomendação H.323 [5, 10, 17, 20, 21, 33, 34, 35, 36] é um conjunto de protocolos para comunicação de dados, áudio e vídeo, sobre uma rede IP, como a Internet. A norma H.323 foi aprovada em 1996 pelo grupo de estudo 16 (SG16) da ITU. A versão 2 foi aprovada em Janeiro de 1998 e especifica mais funções na área de segurança e serviços suplementares (registro, admissão e *status*). É uma norma da ITU-T (International Telecommunications Union - Teleconferencing). O H.323 executa também controle de chamadas, gerenciamento de multimídia, gerenciamento de largura de banda, bem como as interfaces entre LAN's e outras redes (PSTN).

O protocolo H.323 foi projetado para operar sobre a camada de transporte da rede. Assim, pode ser usado sobre qualquer rede de pacotes, como Ethernet, TCP/UDP/IP, ATM e *Frame Relay*, para prover comunicação multimídia. Existem outras recomendações da série, como H.310, para conferência em banda larga (B-ISDN), H.320, para conferência banda estreita (N-ISDN), H.321 para conferência em ATM, H.322 para comunicações sobre LAN's que não possuem qualidade de serviço garantida (QoS), e H.324 para conferência em redes PSTN.

2.1. Importância do H.323

A norma H.323 é além de compreensível, flexível e por essas razões e outras razões é aplicada no mercado:

- Define normas para a infra-estrutura existente (ex: redes IP). Foi concebida para compensar as altas variações de latência na LAN e permite que os clientes utilizem aplicações de multimídia sem alterarem a infra-estrutura da rede;
- Ao providenciar a interoperabilidade entre diferentes estruturas, o H.323 permite que os produtos dos usuários possam interoperar com outros produtos H.323;
- H.323 provê normas para a interoperabilidade entre LAN's e outras redes;
- Pode-se restringir a largura de banda disponível para conversação; o suporte multicast reduz os requerimentos de largura de banda;
- H.323 tem o apoio de várias companhias e organizações de computação e comunicações, incluindo Intel, Microsoft, Cisco e IBM. O esforço destas companhias gerou um maior nível de atenção no mercado.

2.2. Vantagens do H.323

2.2.1. Normas Codec

O H.323 estabelece normas para compressão e descompressão de fluxo de dados em áudio e vídeo, assegurando que equipamentos de diferentes fabricantes tenham área de apoio comum.

2.2.2. Interoperabilidade

Os usuários conversam sem a preocupação com a compatibilidade do outro terminal. Para assegurar que este possa decodificar a informação, o H.323 estabelece métodos para que os terminais possam se comunicar. A norma também estabelece um *setup* de chamadas e protocolos de controle.

2.2.3. Independência da rede

O H.323 foi concebido para utilizar diferentes arquiteturas de redes. À medida que a tecnologia da rede evolui, as técnicas de gerenciamento melhoram e as soluções H.323 se destacam cada vez mais.

2.2.4. Independência da aplicação e plataforma

A norma não está ligada a qualquer *hardware* ou sistema operacional. As plataformas H.323 estão disponíveis em vários tamanhos e formatos, incluindo *desktop* com vídeo, telefones IP, caixas para TV a cabo, etc.

2.2.5. Suporte multiponto

O H.323 pode suportar três ou mais pontos terminais sem requerer a unidade de controle multiponto, o MCU, que providencia uma arquitetura mais potente e flexível para prover conversações multiponto.

2.2.6. Gerenciamento de largura de banda

O tráfego de áudio e vídeo tem uma largura de banda considerável e pode sobrecarregar a rede. O H.323 resolve esta questão providenciando gerenciamento de largura de banda. Os gerentes da rede podem limitar o número de conexões H.323 simultaneamente dentro da sua rede, ou então a quantidade de largura de banda disponível para as aplicações H.323. Estes limites asseguram que o trafego não será perturbado.

2.2.7. Suporte Multicast

O H.323 suporta o transporte multicast em conversações multiponto. O multicast envia um único pacote para alguns destinos da rede sem réplica; o unicast envia múltiplas transmissões ponto a ponto e o broadcast envia para todos os destinos. Tanto no broadcast como unicast, a rede é ineficientemente utilizada, à medida que os pacotes são duplicados na rede. As transmissões multicast utilizam mais eficientemente a largura de banda, pois todas as estações no grupo multicast têm apenas um único fluxo de dados.

2.2.8. Flexibilidade

Uma conversação H.323 pode incluir terminais com diferentes capacidades. Por exemplo, um terminal que tenha apenas capacidade de áudio pode participar em conversações com terminais que tenham capacidades de áudio,vídeo e/ou dados.

2.2.9. Conversações na Internet

Muitos usuários conversam de uma LAN para uma posição remota. O H.323 estabelece, por exemplo, uma maneira de ligar a LAN com os sistemas ISDN.

2.3. Estrutura

A norma H.323 trata dos documentos técnicos para serviços de comunicação de áudio para LAN's que não provêem qualidade de serviço garantida. No entanto não especifica qual LAN a utilizar, ou qual a camada de transporte deve ser utilizada para conectar várias LAN's, somente os elementos necessários para a interação com redes de comutação de circuitos. A figura 2.1 a seguir mostra os elementos básicos definidos pela norma. Os quatro componentes gerais são:

- **Terminal:** entidade que provê comunicação bidirecional, em tempo real, com outras entidades H.323 (outro terminal, *gateway* ou MCU). Suporta comunicação de voz e vídeo, dados são opcionais;
- ***Gateway*:** executa translações entre vídeo, áudio, e dados. Sua função genérica é refletir as características dos terminais da rede em uma rede de circuitos comutados e vice-versa;

- ***Gatekeeper*:** equipamento que provê conversão de endereços e controle de acesso para terminais H.323;
- **Unidade de Controle Multiponto (MCU):** terminal da rede que tem a capacidade de interligar três ou mais terminais e Gateways para participarem de uma conferência multiponto.

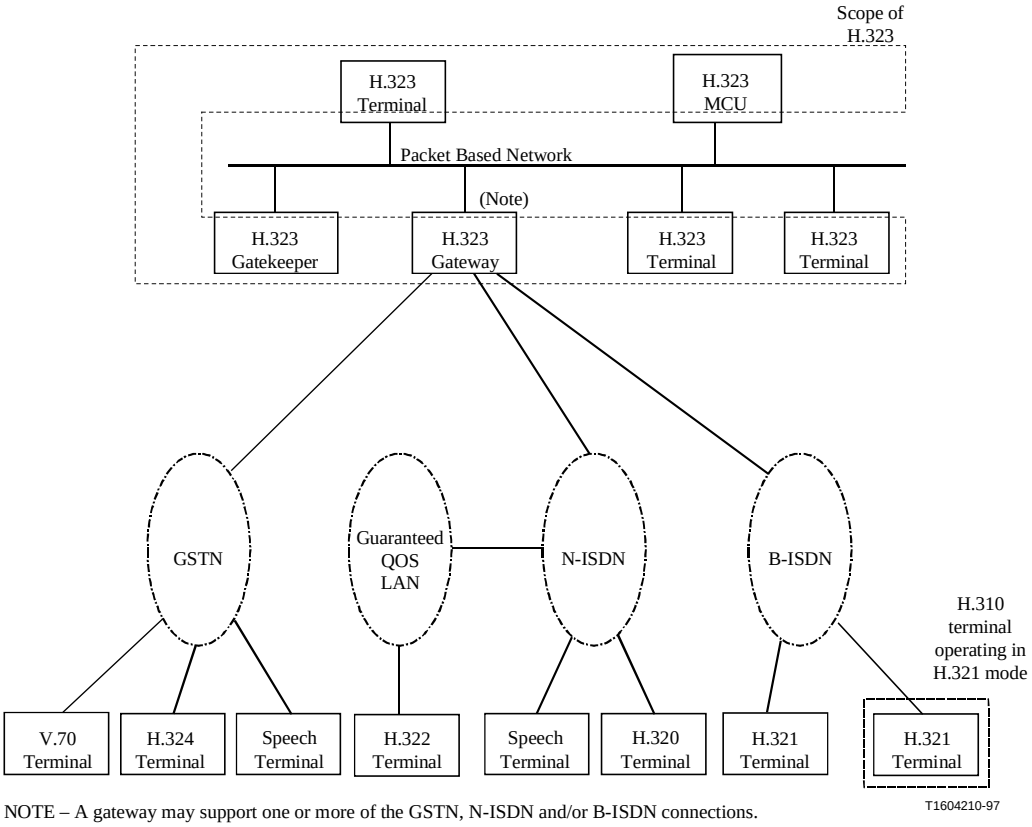

Figura 2.1 - Ambiente do H.323 [35]

2.4. Endereçamento

2.4.1. Endereços de Rede

Cada entidade H.323 tem pelo menos um endereço de rede. Estes endereços apenas servem para identificar a entidade H.323 na rede. No caso da Internet, este endereço é o endereço IP de cada entidade.

2.4.2. Identificadores TSAP

Para cada endereço de rede, uma entidade H.323 tem inúmeros identificadores TSAP. Estes identificadores permitem a multiplexação de vários canais, utilizando o mesmo endereço de rede. Cada terminal tem um identificador TSAP definido que é fixo e conhecido, como o identificador TSAP do canal de sinalização de chamada, e é utilizado no *setup* da chamada também conhecido como o endereço de transporte de sinalização de chamada.

Nos gatekeepers tem-se um identificador TSAP para multicast e unicast. No unicast tem-se um identificador de TSAP no canal RAS para descobrir o endereço multicast (estes identificadores estão definidos no Apêndice IV da norma H.225).

As entidades H.323 devem utilizar identificadores TSAP dinâmicos para os canais H.245 e os canais de áudio e vídeo. O gatekeeper deve utilizar identificadores TSAP dinâmicos para canais de sinalização de chamadas.

Os canais RAS e de sinalização devem ser redirecionados para identificadores dinâmicos de TSAP durante o registro.

2.4.3. Endereços *Aliases*

Um terminal pode ter um ou mais endereços associados. Os endereços *Aliases* podem representar um terminal, ou podem representar uma conferência que a unidade de controle multiponto está executando. Este tipos de endereços providenciam um método alternativo de endereçamento de um terminal, com endereços mais legíveis para o usuário. Os endereços *Aliases* devem ser únicos dentro de uma zona. Os gatekeepers, MC's, MP's não têm endereços *Aliases*.

2.5. Protocolos e funcionamento

2.5.1. Pilha de Protocolos

A norma H.323 define uma pilha de protocolos e está descrito na figura 2.2 a seguir. Os protocolos especificados na pilha foram colocados sobre o modelo TCP/IP, modelo utilizado pela Internet (para a norma H.323 não é necessário a utilização do TCP/IP como suporte).

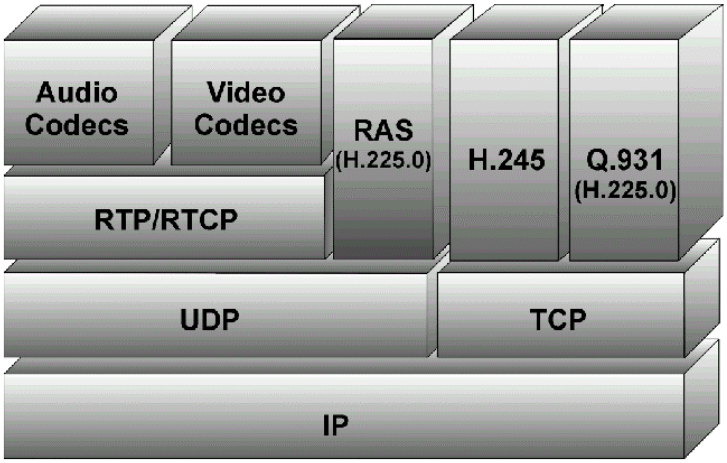

Figura 2.2 – Pilha de protocolos H.323 [18]

O H.323 requer conexões confiáveis (TCP) para funções de controle nas quais é necessário garantir sequência sem erros e fluxo controlado de pacotes (o processo pode atrasar a transmissão e reduzir a qualidade). Para funções de envio de áudio e vídeo, onde o tempo é uma questão essencial, não se pode utilizar protocolos confiáveis. No modelo TCP/IP, para implementar a função de envio de áudio e vídeo, é utilizado o protocolo UDP, que fornece um serviço não confiável, onde a perda de pacotes é possível. Isto não implica que forneça um serviço não funcional em transporte de áudio e vídeo, apenas diminui a qualidade de serviço. O UDP oferece um controle mínimo, e aplica a idéia do *best effort*. Os protocolos de controle e troca de informação são especificados na tabela 2.1.

Tabela 2.1 - Protocolos de controle e informação

Ação	**Protocolo H.323**	**Protocolo de Transporte**
Requisição por partes dos terminais ao gatekeeper, permissão e largura de banda para iniciar uma sessão H.323	H.225 RAS	UDP
Negociação entre os terminais e estabelecimento de setup de chamada	Q.931	TCP
Troca de capacidades entre terminais e setup RTP	H.245	TCP
Troca de áudio/vídeo entre terminais	RTP/RTCP	UDP

Na pilha de protocolos, o protocolo RTP não está a nível de protocolos de transporte, mas é considerado um protocolo de transporte.

2.5.2. Canal RAS

O canal RAS deve ser usado para transportar mensagens com a finalidade de descobrir o gatekeeper correspondente e o processo de registro de um terminal. Então associa um endereço *Alias* de um ponto terminal ao seu endereço de transporte de sinalização de chamada que é utilizado no *setup*. O canal RAS é um canal não confiável. Devido a isso, a norma H.225 recomenda que existam contadores de *time out* de registros de tentativas para se ter controle de progresso do registro.

2.5.2.1. Descoberta do gatekeeper

Existem dois métodos para um terminal encontrar o gatekeeper, o método manual e o método automático.

No método manual o administrador de rede tem que colocar no diretório de inicialização o endereço de transporte do canal RAS do gatekeeper correspondente a ele para que possa implementar o registro.

Já no método automático, o terminal deve enviar uma mensagem *multicast* procurando por seu gatekeeper. Esta mensagem é enviada para o endereço *standard multicast* da descoberta do gatekeeper. Um ou mais gatekeepers podem responder com resposta afirmativa (GCF) retornando o endereço do seu canal RAS. Se mais de um gatekeeper responder, o terminal deve escolher qual o gatekeeper utilizará. Se nenhum gatekeeper responder dentro de cinco segundos após o primeiro pedido, ele reenvia outro pedido. Se após este procedimento não obter resposta, o terminal deve usar o método manual de descoberta do gatekeeper. A descoberta do gatekeeper deve ser feita periodicamente (cada vez que o terminal se liga), por isso o gatekeeper deve estar preparado para pedidos múltiplos.

2.5.2.2. Registro de um terminal

O registro de um terminal ao gatekeeper informa a este o seu endereço de transporte e endereço *Alias* ficando assim em uma zona. Este processo de registro deve fazer parte da configuração do terminal logo a seguir. A descoberta do gatekeeper utiliza o endereço do canal RAS que lhe foi atribuído, conforme a figura 2.3 a seguir:

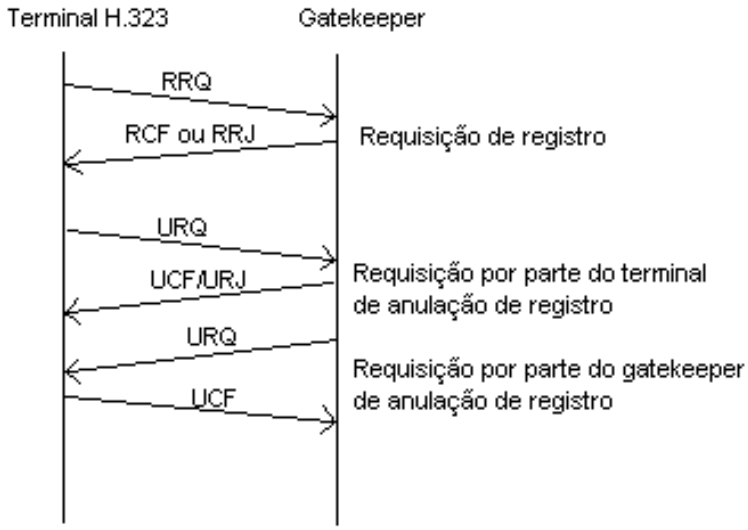

Figura 2.3 - Requisição de registro de um terminal a um gatekeeper [35]

As mensagens utilizadas para registro são as mensagens RRQ (Registration Request) e, devem receber mensagens RCF/RRJ se a confirmação foi bem sucedida ou não foi bem sucedida respectivamente. Para que um terminal possa mudar o seu endereço Alias ou seu endereço de transporte existe um processo de retirar o registro, URQ (anulação de registro).

Podem existir terminais não registrados. Estes terminais não podem participar de processos de controle de admissão (ARQ/ACF), e não usufruem de funções do gatekeeper, como tradução de endereços e controle de largura de banda.

2.5.2.3. Localização de um terminal

Quando um terminal, ou gatekeeper, tem endereço *Alias* do terminal que se quer conectar, e quer saber o endereço do canal de sinalização de chamada destino, o terminal ou gatekeeper deve enviar uma mensagem *multicast* de requisição de localização, para o endereço *standard* de descoberta *multicast*. O gatekeeper que tem o terminal registrado, deve responder com uma mensagem de confirmação de localização, contendo o endereço do canal de sinalização de chamada do terminal ou gatekeeper.

Para localizar terminais que estejam fora da Internet, localizados na PSTN, o gateway que faz a interface com estes terminais deve ter vários endereços *Aliases* que os relacionem com o telefone (este procedimento está fora da norma H.323 e é implementado de acordo com o fabricante).

2.5.2.4. Processo de admissão e determinação de largura de banda

Quando um terminal quer realizar uma chamada é utilizado o processo de admissão (ARQ/ACF). Com este processo consegue-se implementar o esquema de segurança. Quando o terminal envia a mensagem de admissão (ARQ), ele especifica a largura de banda que necessita. Esta largura de banda tem o valor limite máximo para taxa de bit nos canais de transmissão e recepção RTP, excluindo todos os cabeçalhos do RTP e dos restantes protocolos da rede.

Este esquema pode servir para o gatekeeper não deixar realizar uma chamada, se verificar que não existe largura de banda disponível (este esquema não garante a qualidade de serviço, porque a largura de banda não é garantida, dependendo da utilização da rede). Durante uma chamada podem existir chamadas de troca de mensagens BRQ (requisição de largura de banda). Por exemplo, se um gatekeeper verificar que a rede está ficando muito sobrecarregada, pode pedir ao terminal que diminua a largura de banda que está utilizando, implicando até em troca de codec utilizado para um de maior compressão. Neste processo pode-se ter funções para verificar se o terminal está registrado (status) e funções para acabar com a admissão (disengage). Se um terminal que está efetuando uma chamada inicia o processo de admissão, abrindo um canal RAS, mas ao verificar que o terminal destino está indisponível utiliza funções disengage para fechar o canal RAS.

2.5.3. Canal de sinalização de chamada (canal Q.931)

Este é o canal usado para transportar mensagens Q.931 no *setup* da chamada. Os terminais que não estão registrados devem conter nos seus registros os endereços do canal de sinalização de chamada dos terminais que vão comunicar. Os terminais que estão registrados no gatekeeper vão utilizar este para que, através do endereço *Alias*, encontre os endereços de transporte do canal de sinalização de chamada correspondente. Esses canais de sinalização de chamada podem ser repartidos em dois tipos:

- Sinalização de chamada direta;
- Sinalização de chamada encaminhada.

2.5.3.1. Sinalização de chamada direta:

Os terminais apenas executam as funções do canal RAS com o seu gatekeeper correspondente e os gatekeepers comunicam-se entre eles para obter o endereço de transporte do canal de sinalização de chamada. Com esse endereço, os terminais comunicam-se diretamente entre eles para fazer o setup Q.931. A figura 2.4 ilustra a chamada direta entre dois terminais.

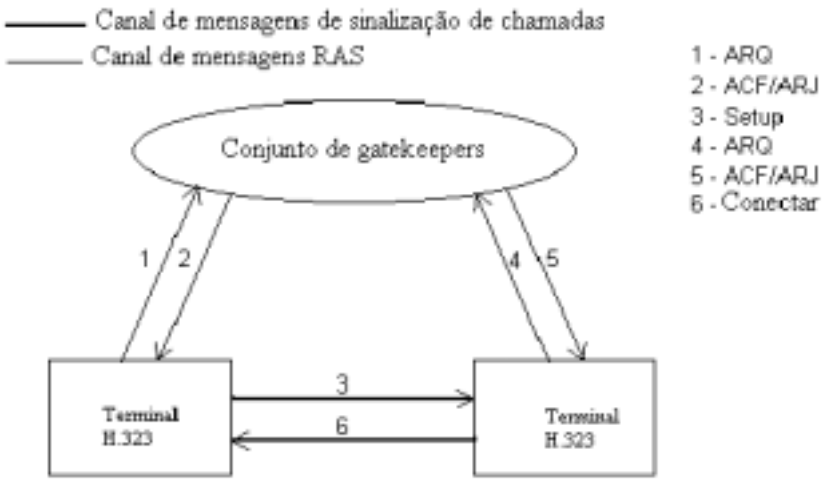

Figura 2.4 - Estabelecimento do canal de sinalização de chamada diretamente entre os dois terminais [35]

2.5.3.2. Sinalização de chamada encaminhada:

Na sinalização de chamada encaminhada, o canal de sinalização de chamada não é estabelecido diretamente entre os terminais. Ele é encaminhado entre o conjunto de gatekeepers existentes na rede para chegar da origem ao destino. Esta hipótese de encaminhamento do canal de sinalização Q.931 também pode ser aplicada ao canal de controle H.245 como se pode ver na figura 2.5 a seguir.

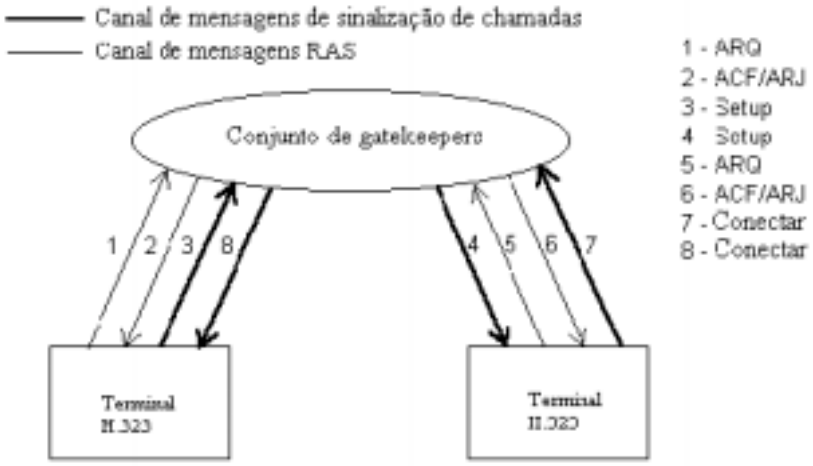

Figura 2.5 - Estabelecimento do canal de sinalização de chamada encaminhado através do gatekeeper [35]

2.5.4. *Setup* de chamada(Q.931)

O *setup* da chamada entre dois terminais utilizando o protocolo Q.931 (referido na norma H.225) especifica procedimentos dependendo da localização física dos terminais e o registro no gatekeeper. Os casos abaixo descrevem as possibilidades para este item.

2.5.4.1. Chamada Básica (nenhum terminal registrado no gatekeeper)

Os dois terminais comunicam diretamente depois de aberta uma conexão TCP. Eis o procedimento:

1. Terminal 1 envia uma mensagem de *setup* para o identificador TSAP do canal de sinalização de chamada;
2. Terminal 2 responde com uma mensagem de conexão contendo o endereço do canal de transporte de controle H.245 para efetuar as próximas operações.

A figura 2.6 ilustra esta chamada.

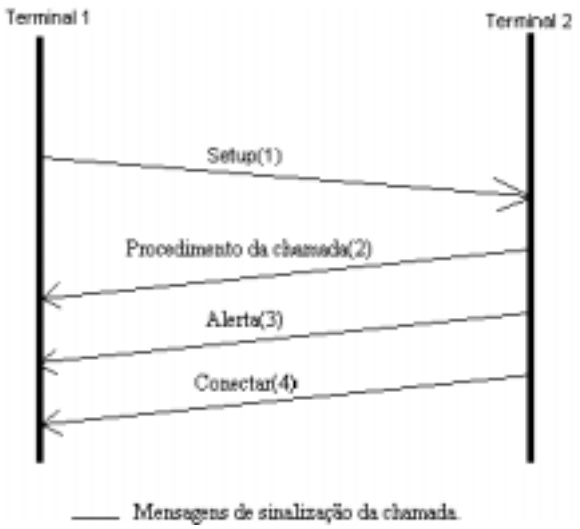

Figura 2.6 - *Setup* da chamada direta entre dois terminais [35]

2.5.4.2. Os dois terminais estão registrados no mesmo gatekeeper

Neste caso pode ser usado dois métodos: a sinalização chamada direta e sinalização de chamada encaminhada.

2.5.4.2.1. Sinalização de chamada direta

1. O terminal 1 inicia a troca da ARQ/ACF(processo de autorização) com o gatekeeper. O gatekeeper deve responder com o endereço do canal de transporte do setup de chamada do terminal 2;
2. O terminal 1, utilizando o canal de transporte, envia mensagens de setup ao terminal 2;
3. Se o terminal 2 aceitar a chamada, inicia a troca ARQ/ACF com o gatekeeper;
4. O terminal 2 responde com mensagens de conexão que contêm o endereço do canal de transporte do controle H.245.

Esta sinalização é mostrada na figura 2.7.

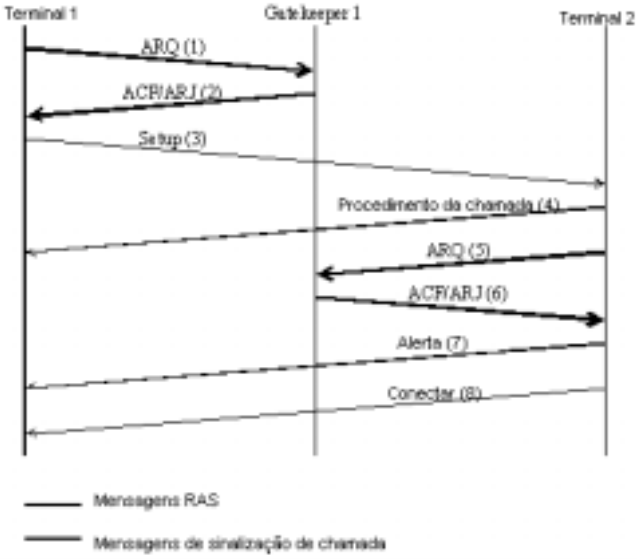

Figura 2.7 - *Setup* da chamada quando os dois terminais estão registrados no mesmo gatekeeper (chamada direta) [35]

2.5.4.2.2. Sinalização de chamada encaminhada

Neste caso, o procedimento é idêntico, apenas o canal de transporte não é diretamente feito com o terminal 2, mas é feito com o gatekeeper, como é mostrado na figura 2.8.

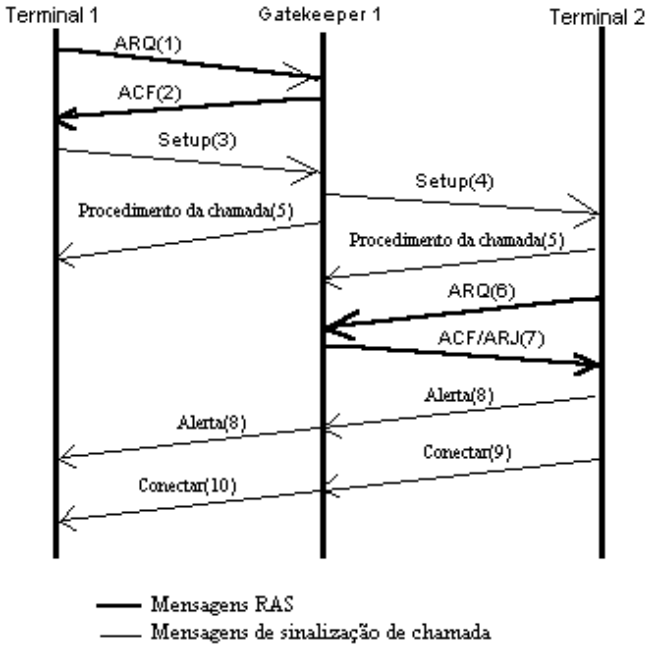

Figura 2.8 - *Setup* da chamada quando os dois terminais estão registrados no mesmo gatekeeper (chamada encaminhada) [35]

2.5.4.3. Apenas o terminal de origem está registrado no gatekeeper

Neste caso, o terminal 1 escolhe sinalização de chamada direta, conforme a figura 2.9.

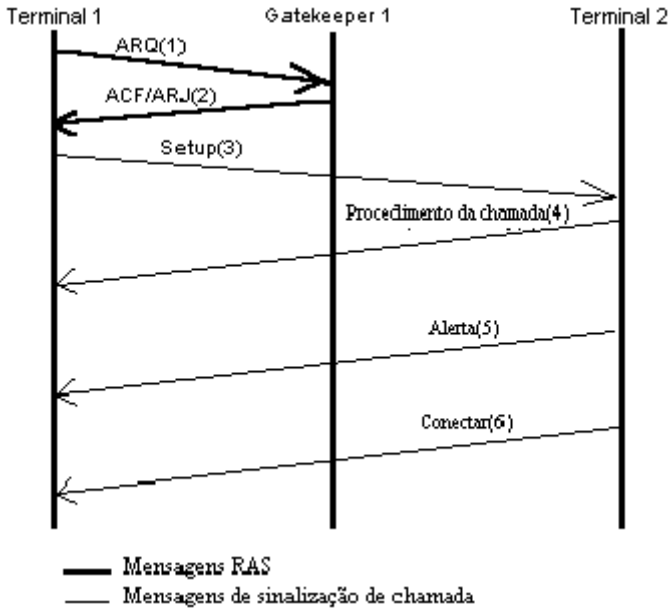

Figura 2.9 - *Setup* da chamada quando apenas o terminal de origem está registrado no gatekeeper [35]

1. O terminal 1 inicia uma troca ARQ/ACF com o gatekeeper, o qual retorna o endereço do canal de transporte de sinalização de chamada; este vai ser utilizado para realizar o *setup* de uma chamada básica entre o terminal 1 e o terminal 2;
2. Utilizando sinalização de chamada encaminhada o procedimento é idêntico, mas a sinalização de chamada é enviada pelo gatekeeper ,como no caso anterior.

2.5.4.4. Apenas o terminal 2 está registrado no gatekeeper

Se o gatekeeper escolher sinalização de chamada direta:

1. O terminal 1 envia uma mensagem de *setup* para o endereço *standard* do canal de transporte de sinalização de chamada do terminal 2;
2. O terminal 2 aceita a chamada e realiza o procedimento de autorização ARQ/ACF com o gatekeeper;
3. O terminal 2 responde com uma mensagem de conexão que contém um endereço do canal de transporte controle H.245, o qual é utilizado a seguir. É possível que a autorização seja falhada (tenha recebido uma mensagem ARJ). Nesse caso é enviado uma mensagem de erro para o terminal 1.

No caso da sinalização de chamada encaminhada, pode-se ver o seu funcionamento de acordo com a figura. No caso de uma autorização não obter sucesso, é necessário que o terminal oposto realize uma operação de liberação de recursos do seu gatekeeper, para que este dê como terminada a conexão (DRQ/DCF, *Disengage Request, Disengage Config*). A figura 2.10 ilustra a situação.

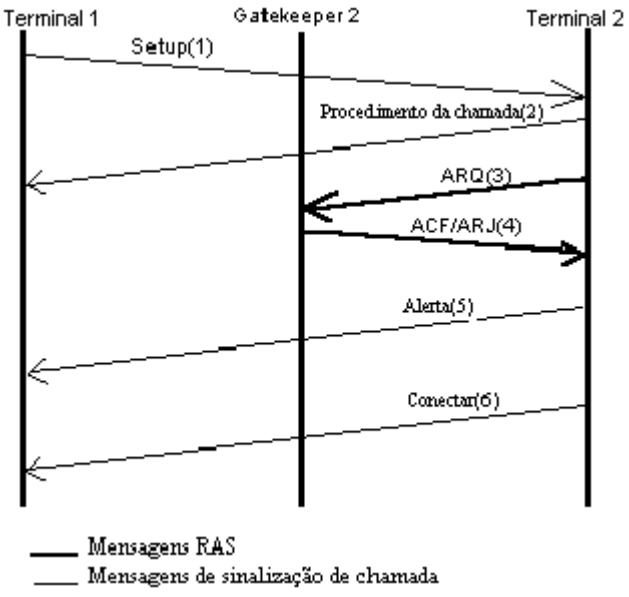

Figura 2.10 - *Setup* da chamada onde o terminal 2 está registrado no gatekeeper [35]

2.5.4.5. *Setup* de chamada via gateway

2.5.4.5.1. *Setup* de chamada com a chamada a dar entrada no gateway

Quando um terminal externo (telefone normal PSTN ou N-ISDN) à Internet, liga a um gateway, o *setup* da chamada entre o gateway e o terminal destino da Internet é idêntico ao comentado anteriormente. O gateway deve enviar mensagens do procedimento das chamadas ao telefone para que este esteja informado do estabelecimento da chamada. O gateway realiza dois estados de chamada. O estado onde aceita o número de telefone (proveniente de um telefone), se for ISDN tem que aceitar número SDE, se for um telefone normal (apenas de fala) deve aceitar números DTMF, para identificações do usuário, indicações essas que vão servir para realizar o segundo estado da chamada onde esse número vai ser utilizado para conectar ao terminal destino na rede.

2.5.4.5.2. Quando o *setup* da chamada é para fora do gateway

Quando um terminal quer ligar para um terminal externo via gateway , o *setup* da chamada é idêntico ao *setup* básico (entre o terminal e o gateway). O gateway deve receber o número de telefone de destino, o qual vai ser utilizado na chamada exterior. Ao realizar esta chamada ele deve informar ao terminal origem, dentro de 4 segundos, como está decorrendo a realização da chamada (mensagens de progresso) .

2.5.4.6. *Setup* da chamada com o MCU

Neste caso todos os terminais trocam sinalização de chamada com o MCU. O *setup* entre um terminal e o MCU tem o mesmo procedimento dos *setup* de chamadas vistos anteriormente entre dois terminais.

2.5.5. Controle das chamadas (H.245)

Depois de realizar o *setup* de chamadas, como comentado anteriormente utilizando Q.931, e de receber o endereço de transporte do canal de controle H.245, os terminais irão trocar funções de controle. Essas funções de controle podem ser resumidas em:

- Troca de capacidades;
- Abertura de canais lógicos para transporte de áudio e vídeo (canais RTP);
- Fechamento de canais lógicos por parte do terminal que recebe a chamada;
- Requisição dos modos de áudio e vídeo;

- Determinação dos tempos que demora percorrendo o caminho entre os dois terminais (*Round time delay*);
- Determinação do mestre/escravo.

2.5.5.1. Troca de capacidades

Neste caso, os terminais irão trocar entre eles as suas capacidades (ex: terminal pode ter três capacidades de compressão/descompressão de áudio G.711, G.723, G.729), isto é, qual a capacidade de cada terminal de receber e decodificar a informação proveniente do outro. Não é necessário que o terminal entenda todas as capacidades provenientes do terminal origem; cada terminal envia o seu conjunto de capacidades aos outros, em seguida é determinado o conjunto de capacidades comum a todos os pontos terminais envolvidos.

Com isto, cada terminal sabe que tipo de informação pode enviar ao outro, em qualquer momento dentro da transmissão (codec's) para outro contido no conjunto de capacidades comuns.

2.5.5.2. Abertura de canais lógicos

Depois de cada terminal conhecer as capacidades do outro, tem-se que abrir canais lógicos, pelos quais vão ser transportados os fluxos dos bits de áudio e vídeo. Neste caso são canais lógicos RTP (canais lógicos utilizados para transportar informação em tempo real).

Cada canal lógico é unidirecional. Com isto, para obter um canal *full duplex* entre dois terminais, tem-se que abrir dois canais RTP. A cada canal lógico RTP está associado um canal RTCP o qual implementa o controle nos canais RTP.

2.5.5.3. Fechamento de canais lógicos por parte do terminal que recebe a chamada

Normalmente um canal lógico RTP é aberto e fechado do lado do emissor. É definido um mecanismo que possibilita que o receptor requisite fecho do canal lógico que está chegando, possibilitando uma gestão de recursos do receptor. Se o emissor está tentando abrir um canal que o receptor não consegue decodificar ou não tem mais recursos para receber, o receptor pode pedir o fecho desse canal.

2.5.5.4. Requisição dos modos de áudio e vídeo

Depois dos terminais trocarem um conjunto de capacidades, são estabelecidos os codecs de áudio e vídeo a serem utilizados.

2.5.5.5. Determinação do tempo que demora a percorrer o caminho entre dois terminais

Este item é especificado na norma H.245. Esse mecanismo determina o tempo que o pacote demora a ir do emissor ao receptor. A determinação desse tempo vai ser essencial para o protocolo RTCP.

2.5.5.6. Determinação do Mestre/Escravo

Podem haver conflitos quando dois terminais estão envolvidos numa iniciação de chamada, onde estas representam eventos iguais. Quando dois terminais querem se conectar um ao outro simultaneamente, um mecanismo vai determinar qual dos dois é o mestre e qual é escravo, ou seja, qual é o que inicia a chamada e qual é que a recebe.

2.5.6. Resumo do funcionamento dos protocolos

Quando um terminal se liga à rede ele executa a procura do gatekeeper e procede ao seu registro atribuindo-lhe os endereços Aliases e os endereços de transporte através do canal RAS.

Quando um terminal quer comunicar com outro ele abre um canal RAS com o seu gatekeeper, onde será executada funções de admissão e é retornado o endereço de transporte do canal de sinalização de chamada Q.931. Em seguida é utilizado o protocolo Q.931 para fazer o setup da chamada. Ao fazer este setup obtém-se o endereço do canal de controle H.245. Com este endereço abre-se um canal H.245 que faz a negociação das capacidades, e abre-se os canais RTP e RTCP correspondentes a cada um, entre emissor e o receptor e vice versa.

Com os canais RTP e RTCP abertos, o fluxo de bits provenientes da fonte são comprimidos através de um codec, e em seguida empacotados em pacotes RTP para serem enviados ao receptor (o processo é invertido). Basicamente no canal RTCP é transportada a informação para monitorar a qualidade de serviço. Pode-se ver esta sequência na figura 2.11 a seguir.

Para garantir uma determinada qualidade de serviço utiliza-se o protocolo RSVP para reservar um determinada largura de banda na Internet necessária para os canais RTP.

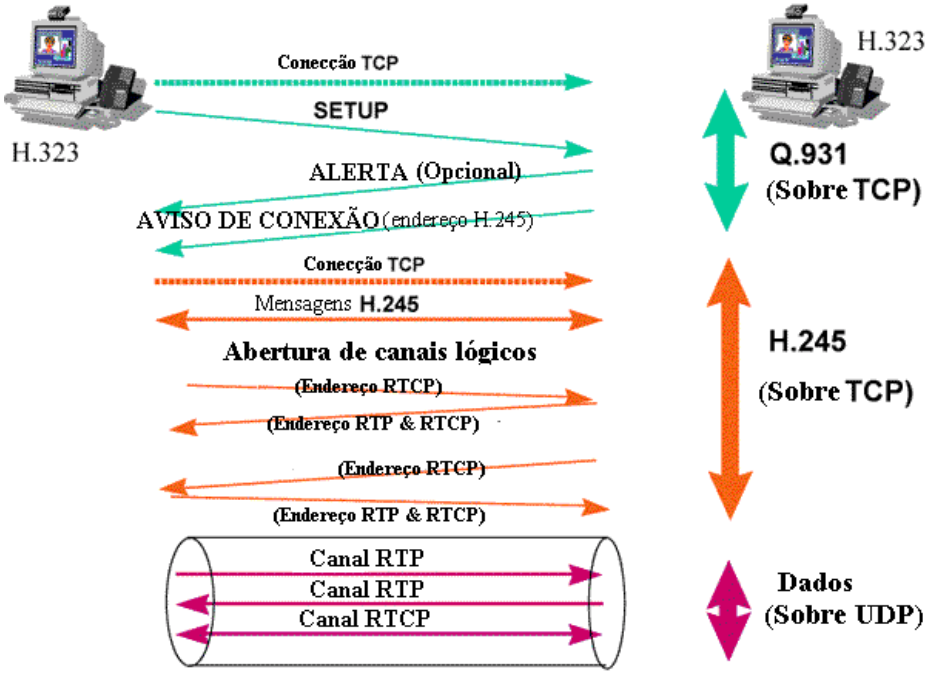

Figura 2.11 - Diagrama de utilização dos protocolos [19]

3. Componentes

3.1. Terminal

O terminal [6, 10, 13, 16, 20, 30, 35] provê comunicação bidirecional, em tempo real, com outras entidades H.323 (outro terminal, *gateway* ou MCU). Para obter isto, o terminal deve suportar comunicação de voz e vídeo, dados são opcionais, podendo ser um equipamento telefônico com a funcionalidade de comunicação através do protocolo IP, um microcomputador com *software* específico e *hardware* multimídia para comunicação de voz. Outros tipos de equipamentos que possam estar conectado em rede e permitam, no mínimo, a comunicação de voz entre usuários são aceitáveis.

Os terminais devem ser capazes de codificar áudio para transmissão e decodificar os sinais recebidos. Além disso, devem suportar a funcionalidade de sinalização na unidade de controle do sistema. Tais funcionalidades consistem em:

- Controle H.245;
- Controle de Chamada;
- Sinalização RAS.

A figura 3.1 abaixo ilustra os elementos internos de um terminal H.323.

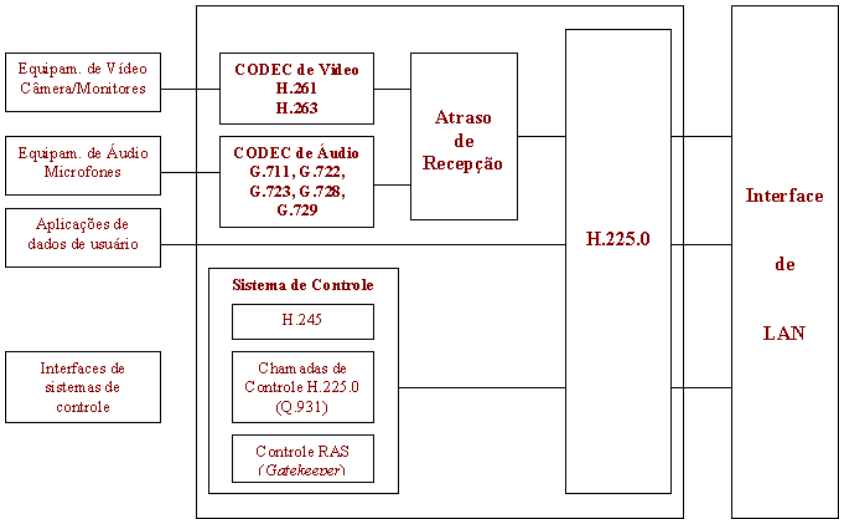

Figura 3.1 - Terminal H.323 [35]

Estes elementos são os seguintes:

- **Codec (codificador/decodificador) de vídeo:** suporta obrigatoriamente a Recomendação H.261 e opcionalmente a H.263, referentes à codificação de sinal de vídeo. A implementação deste elemento é opcional.
- **Codec de áudio:** suporta obrigatoriamente a Recomendação G.711 e opcionalmente as Recomendações G.722 (sinal de 7kHz codificado em 64kbps), G.728, G.722, áudio MPEG1 e G.723.1. A implementação deste elemento é obrigatória.
- **Atraso de recepção:** responsável pelo controle do *jitter* nas recepções de áudio e/ou vídeo. A implementação deste elemento é obrigatória.
- **Canal de dados:** apesar de não estar explicitamente indicado na figura, suporta um ou mais canais de dados, uni ou bidirecionais. O T.120 é a base padrão para interoperabilidade entre terminais H.323 ou outro tipo suportado pelos ambientes externos ligados por *gateways*. A implementação deste elemento é opcional.
- **Controle H.245:** esta função provê ao terminal a habilidade de enviar mensagens para negociar o uso e a capacidade do canal. Mensagens H.245 encontram-se dentro de quatro categorias: *Request*, *Response*, *Command* e *Indication*. Esta função também é usada para abrir e encerrar um canal lógico entre dois terminais. Quando um canal é aberto, um número único de canal lógico é associado e este canal e pode então ser usado para comunicar a capacidade dos dois terminais. Esta troca de capacidade permite aos terminais identificarem um método compatível para a transmissão da informação (voz, vídeo e dados).
- **Controle RAS (*Registration, Admission and Status*):** provê controle de mensagens de sinalização para registro, admissão, troca de banda, pedido de informação sobre a situação atual e desvinculação com o *gatekeeper*. O canal RAS é aberto entre os terminais e o *gatekeeper* antes que qualquer outro canal entre pontos H.323 sejam configurados.
- **Controle de Chamadas:** usado para a sinalização e configuração de chamadas entre dois pontos H.323. O canal de sinalização de chamada é aberto antes do estabelecimento do canal H.245.
- **H.225.0:** estabelece o formato das mensagens trocadas pelos canais lógicos de áudio, vídeo, dados ou controle.

3.1.1. Softwares para Videoconferência

Existem alguns software capazes de realizar uma videoconferência. Entre eles, pode-se listar os seguintes:

3.1.1.1. Microsoft NetMeeting

Apresenta-se como uma ferramenta com disponibilidade de áudio e vídeo, bate-papo, compartilhamento de programas, quadro de comunicações, transferência de arquivos, chamadas avançadas e segurança. A conexão dos usuários é feita através de um servidor ils ou através de um número IP de um dos usuários.

O NetMeeting tem sido indicado como o mais adequado para a realização de videoconferência, pela capacidade de enviar e receber imagens de vídeo em tempo real, usando equipamentos compatíveis com o Windows, como uma câmera de vídeo. Possui um mecanismo de ajuste da sensibilidade do microfone possibilitando uma melhor qualidade do som recebido pelo computador remoto.

Apresenta-se com a possibilidade de localizar pessoas conectadas a um determinado diretório, através da listagem dos usuários que se encontram numa janela intitulada “Encontrar Alguém”.

No recurso de compartilhamento de arquivos, apresenta-se como um recurso que permite a outro computador remoto ter acesso a um documento que é disponibilizado, mas, não é de muito fácil interação quanto a escrita neste documento. Esta se torna extremamente lenta para aquele que está acessando e não disponibilizando.

Quanto ao envio e recepção de imagem, esta pode ser boa qualidade sendo que o NetMeeting permite alterar o tamanho da janela de vídeo que está sendo enviado, mas só é possível o envio e recebimento de imagens entre os dois primeiros que se conectaram no diretório. Os demais contentam-se apenas com a interação via chat, quadro de comunicação ou compartilhamento de arquivos. A conferência pode chegar somente até o número de 8 participantes.

Quando se necessita aumentar o número de conferencistas, é necessário o uso de um MCU para fazer esta conferência multiponto, ampliando em muito o número de pessoas que podem trocar vídeo e áudio. Outra alternativa, é a utilização de um refletor do tipo MeetingPoint, que gerencia a videoconferência [31].

3.1.1.2. CuSeeMe

Semelhante ao NetMeeting, permite a conexão de várias pessoas ao mesmo tempo, com recursos de áudio, vídeo, chat e compartilhamento de dados. Necessita de um servidor ils. O bate-papo é simplificado se comparado com o do NetMeeting.

Da mesma empresa do MeetingPoint, o CuSeeMe adapta-se bem ao uso de uma conferência do tipo multiponto quando a conexão se faz através desse refletor.

Também oferece sistema de comunicação segura entre os usuários e consta como um dos mais utilizados em comunicação a distância quando se requer a utilização de áudio e vídeo. Apresenta-se com a possibilidade de conexão de até mais de 12 usuários em uma conferência.

Uma das desvantagens do *software* é a limitação quanto a transferência de arquivos durante a conferência.

3.1.1.3. IVisit

Esta ferramenta propicia a comunicação ponto-a-ponto, necessitando apenas de computador, uma conexão com a internet, o *software* iVisit e uma câmera de vídeo.

Aparece como um dos mais simples e eficientes mecanismo de interação multimídia entre os internautas. Não necessita de um servidor e permite aos conferencistas a troca de vídeo, áudio e mensagem de texto em tempo real.

3.1.1.4. NetworkVideo

Um sistema pioneiro na utilização de Mbone (*backbone* multiponto), mas não usa sinais de áudio e não trata de documentos.

3.1.1.5. IVS

Este sistema envia as imagens (vídeo) por uma porta e os sons (áudio) por outra, de forma que, a partir do envio, desconsidera-se o sincronismo entre as mídias que podem chegar ao destino dessincronizadas. Também aproveita a estrutura MBone para a transmissão multiponto. Permite a realização de conferência segura pois apresenta sistema de segurança.

3.1.1.6. VIC

O sistema objetiva a flexibilidade, combinando diferentes componentes responsáveis cada pela codificação/decodificação de sinais de vídeo, tratamento de ruído, editor gráfico cooperativo, etc. Um outro sistema integra todos os aplicativos num único ambiente. Este também pode não apresentar sincronismo na transmissão de áudio e vídeo.

3.1.1.7. Meeting Point

Permite uma videoconferência ou reunião com número maior de participantes. Oferece serviços de conferências multiponto para clientes baseados no padrão H.323, incluindo o CuSeeMe.

O Meeting Point oferece fácil integração em um ambiente de rede e pode ser acessado pela maioria dos diversos tipos de computadores hoje existentes, via um navegador *Web*. *Softwares* clientes de videoconferência ponto-a-ponto que utilizam as vantagens do padrão H.323 (padrões H.320 são dirigidos à videoconferência em redes comutadas, como a rede digital de serviços integrados. A recomendação H.323 é uma extensão do H.320 para redes locais. O H.323 pode ser aplicado para vídeo na Internet e sessões ponto-a-ponto e multiponto) podem participar em conferência multiponto através da conexão a um servidor como o Meeting Point. Alguns desses *softwares* são: Intel Business Vídeo Conferencing, Intel Team Station, Microsoft NetMeeting e Picture Tel Live Lan, além do CuSeeMe.

3.1.1.8. OhPhone

O ohphone faz parte das bibliotecas do OpenH.323 e será comentado mais adiante. Este é um *software* livre e seu código está aberto a novas implementações. Dentre os software que realizam a função de *front-end* do ohphone, pode-se citar o Gong [35].

3.2. Gatekeeper

Gatekeeper [10, 11, 13, 20, 35] é o componente mais importante de uma rede H.323. Ele atua como ponto central para todas as chamadas dentro de sua zona e provê serviços de controle de chamada para estações registradas. Em muitas implementações, um Gatekeeper H.323 age como um interruptor virtual.

Gatekeepers executam duas funções de controle de chamada importantes. A primeira é tradução de endereço *Alias* dos terminais de rede e gateways para endereços IP ou IPX, como definido na especificação da RAS. A segunda função é a administração de largura de banda, que também é designada dentro da RAS. Por exemplo, se um gerente de rede especificou um limite para o número de conferências simultâneas na rede, o Gatekeeper pode recusar fazer algumas conexões, uma vez que o limite é alcançado. O efeito é controlar a largura de banda total da conferência, para alguma fração do total disponível. A capacidade restante permanece para e-mail, transferência de arquivo, e outras atividades da rede. A coleção de todos os Terminais, Gateways e MCU's

administrados por um único Gatekeeper é conhecida como uma zona H.323 é mostrado na figura 3.2 a seguir.

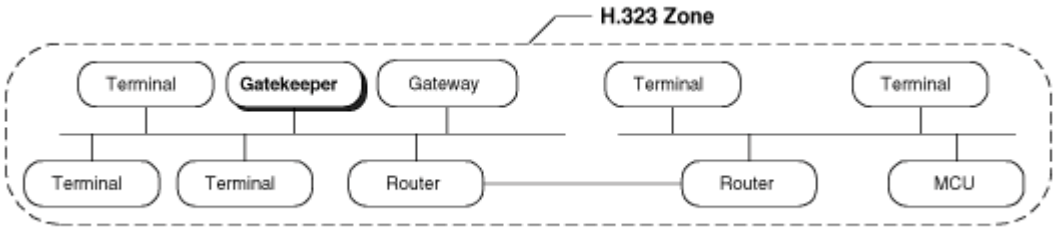

Figura 3.2 – Zona H.323 [18]

Uma opção, mas valiosa característica de um gatekeeper, é sua habilidade para rotear chamadas H.323. Pelo roteamento de uma chamada através de um gatekeeper, um serviço pode ser controlado mais efetivamente. Provedores de Serviço precisam desta habilidade para avisar as chamadas registradas pelas suas redes. Este serviço também pode ser usado para redirecionar uma chamada para outra estação se uma estação chamada está indisponível. Além disso, o gatekeeper é capaz de rotear chamadas H.323 podendo ajudar nas decisões que envolvam balanceamento entre gateways múltiplos. Por exemplo, se uma chamada é roteada por um gatekeeper, este gatekeeper pode redirecionar a chamada para um dos muitos gateways baseados em alguma lógica de roteamento proprietária.

Enquanto um Gatekeeper é logicamente separado das estações H.323, os fabricantes podem incorporar funcionalidades de Gatekeeper na implementação física de Gateways e MCU's.

Um Gatekeeper não é requerido em um sistema H.323. No entanto, se um Gatekeeper está presente, os terminais têm que fazer uso dos serviços oferecido por eles. O canal RAS define estes com endereço de tradução, controle de admissão, controle de largura de banda e administrador de zona.

Gatekeepers também podem representar um papel em conexões multiponto. Para apoiar conferências multiponto, usuários utilizam um Gatekeeper para receber canais de controle H.245 de dois terminais em uma conferência ponto-a-ponto. Quando a conferência troca para multiponto, o gatekeeper pode redirecionar o Canal de Controle H.245 para um Controlador Multiponto, o MC. O Gatekeeper não necessita processar a sinalização H.245; só precisa passar ela entre os terminais ou entre os terminais e o MC.

Redes que contêm Gateways também podem conter um Gatekeeper para traduzir endereços E.164 entrantes em endereços de transporte. Como uma zona é definida por seu Gatekeeper, entidades H.323 que contêm um Gatekeeper interno, exigem um mecanismo para desabilitar a

função interna, de forma que quando há entidades múltiplas H.323 que contêm um Gatekeeper em uma rede, as entidades podem ser configuradas na mesma zona.

3.2.1. Principais Funções do Gatekeeper

- **Tradução de Endereços:** Tradução de Endereço *Alias* para endereços de transporte que usam uma tabela atualizada com mensagens de registro. Também são permitidos outros métodos para atualizar a tabela de tradução.
- **Controle de Admissão:** Autorização de acesso de rede que usa requisição de admissão, mensagens de confirmação e rejeição (ARQ/ARC/ARJ). O acesso a rede pode estar baseado em autorização de chamada, largura de banda, ou algum outro critério. Controle de admissão também pode ser uma função nula que admite todos os pedidos.
- **Controle de Largura de Banda** - Suporte para requisição de largura de banda, mensagens de confirmação e rejeição (BRQ/BCF/BRJ), que pode estar baseado em administração de largura de banda. Controle de largura de banda também pode ser uma função nula que aceita todos os pedidos para mudanças de banda.
- **Gerenciamento de Zona** - O Gatekeeper provê as funções para terminais, MCUs, e Gateways que são registrados na sua Zona de Controle.

- Funções Opcionais do Gatekeeper

- **Sinais de Controle de Chamada** - Em uma conferência ponto-a-ponto, o Gatekeeper pode processar sinais de controle de chamada Q.931. Alternativamente, o Gatekeeper pode enviar diretamente sinais estações de trabalho G.931, para qualquer outra estação.
- **Autorização de Chamada** - O Gatekeeper pode rejeitar uma chamada de um terminal baseado na especificação Q.931. As razões para rejeição podem incluir, mas não limitar, restrições de acesso de/para terminais particulares ou Gateways, restringido o acesso durante certos períodos de tempo. O critério para determinar se uma autorização passa ou falha estão fora do escopo do H.323.
- **Gerenciamento de Largura de Banda** - O Gatekeeper pode rejeitar chamadas de um terminal se ele determinar que a largura de banda não é suficiente. Esta função também opera durante uma chamada ativa se um terminal adicional pede largura de banda. O

critério por determinar se a largura de banda especificada está disponível está fora do escopo do H.323.

- **Gerenciamento de Chamadas** - O Gatekeeper pode manter uma lista de chamadas H.323 contínuas para indicar que um terminal chamado está ocupado ou prover informação para a função de Administração de Largura de banda.

3.3. Gateway

O *gateway* [17, 19, 20, 35] é um elemento opcional em uma conferência H.323. Gateways provêem muitos serviços, onde o mais comum é uma função de tradução entre os terminais de conferência H.323 e outros tipos terminais. Esta função inclui tradução entre transmissão formatada e entre procedimentos de comunicação. Além disso, o gateway também traduz codecs de áudio e vídeo e executa configuração de chamada. A figura 3.3 abaixo mostra um gateway H.323/PSTN.

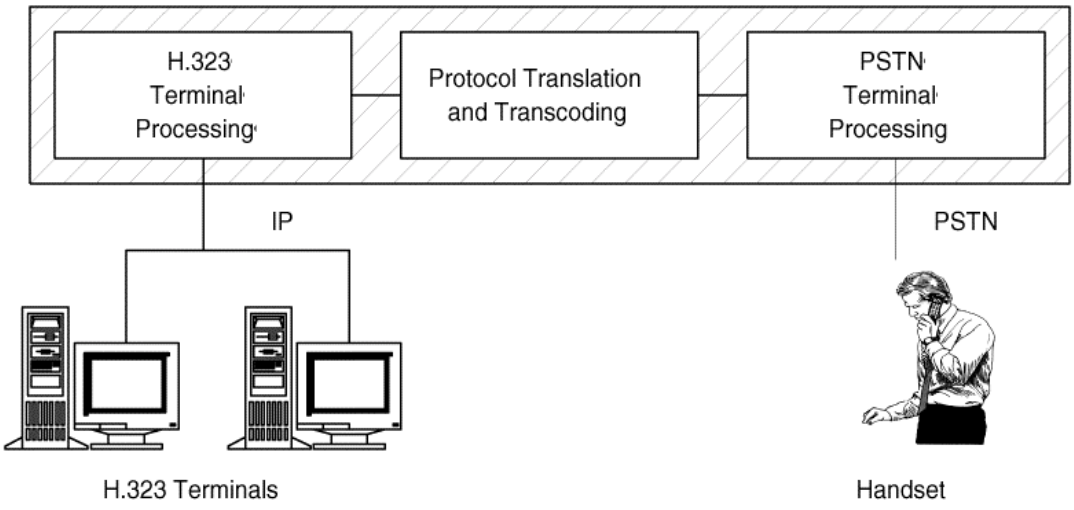

Figura 3.3 – Ligação entre terminais H.323 com a rede de telefonia [18]

Em geral, o propósito do Gateway é refletir as características de uma estação de rede para um estação SCN (Switched Circuit Network) e vice-versa. As aplicações primárias de Gateways são:

- Estabelecer vínculos com terminais de PSTN analógicos;
- Estabelecer vínculos com terminais remotos H.320, através de redes baseadas em ISDN;
- Estabelecer vínculos com terminais remotos H.323, através de redes baseadas em PSTN;

- Estabelecer conexões com linhas digitais de alto velocidade (T1 a 1.554 Mbit/s nos E.U.A ou E1 a 2Mbit/s na Europa), as quais permitem 12 chamadas simultâneas em full duplex;
- Estabelecer conexões com linhas N-ISDN a 64Kbit/s (RDSI faixa estreita) as quais permitem 5 chamadas simultâneas (com largura de banda entre 8-11 Kbps).

Gateways não são requeridos quando não são necessárias conexões com outras redes, isto é, desde que as estações possam se comunicar diretamente com outras estações. Terminais se comunicam com Gateways que usam os protocolos H.245 e Q.931.

Com o transcoder apropriado, Gateways H.323 podem dar suporte a terminais que obedeçam as especificações H.310, H.321, H.322, e V.70.

Muitas funções do Gateway cabem ao gerente determinar. Por exemplo, o número atual de terminais H.323 que podem se comunicar pelo Gateway não está sujeito a padronização. Semelhantemente, o número de conexões de SCN, o número suportado de conferências independentes simultâneas, a função de conversão de áudio/vídeo/dados e inclusão de funções de multipontos pertencem ao fabricante. Incorporando tecnologia de Gateway na especificação H.323, o ITU posicionou o H.323 como a cola que une o mundo de estações de conferência baseado no padrão.

Para obter a interoperabilidade entre a rede IP e a ISDN, ou PSTN, utiliza-se um gateway de voz H.323-ISDN ou um gateway de voz H.323-PSTN. Suas implementações são as seguintes:

- Conversão de áudio em:

ISDN: Se for necessário , porque a ISDN usa G.711;

PSTN: de analógico para G.711.

- Conversão do fluxo de bits:

ISDN: pacotes RTP de/para não empacotado;

GSTN: gerar pacotes RTP.

- Conversão da informação de controle (Gera H.245);
- Conversão do controle de sinalização da chamada;
- Conversão tons DTMF de/para H.245 *userInputIndication* (mensagem de endereços H.245).

3.4. Unidade de Controle Multiponto (MCU)

O MCU (*Multipoint Control Unit*) [18, 21, 35] permite que três ou mais terminais participem de uma conferência audiovisual. Dois ou mais MCU's podem ser interligados em cascata para prover comunicação entre terminais de redes distintas. O MCU provê mescla de áudio e capacidade de troca de vídeo. Este perfil define os requisitos para interatividade de uma conferência multiponto. Transmissões audiovisuais em modo broadcast constitui o escopo deste texto.

Em geral, o MCU obedece os mesmos requisitos dos terminais H.323. Isto inclui as recomendações ITU-T H.221, H.320, H.230 e H.242. Além disso, o MCU ainda obedece aos requisitos da recomendação H.231 que define a representação funcional do MCU. A recomendação H.243, que descreve as especificações detalhadas e procedimentos para uma comunicação entre dois ou mais terminais, também é seguida por esta entidade.

O MCU é constituído por duas partes: o controlador multiponto (MC), que é necessário, e zero ou mais processadores multiponto (MP).

O MC lida com negociações H.245 entre todos os terminais para determinar as capacidades comuns para os processamentos de áudio/vídeo. O MC controla também os recursos da conferência para dizer qual o fluxo de áudio/vídeo deve ser enviado por multicast. O MC não trabalha diretamente com o fluxo de bits de áudio/vídeo, isto é deixado a cargo do MP. O MC e o MP podem existir num componente dedicado ou ser parte de outro componente H.323. Os tipos de conferência suportados pelo MCU podem ser divididos em duas partes: centralizado e descentralizado.

O MP do MCU é responsável pela mistura e comutação dos fluxos de dados áudio/vídeo que chegam das várias fontes. Também é responsável por todo o processamento de dados áudio/vídeo que sejam necessários.

As várias funções e capacidades do MCU são habilitadas e desabilitadas por transmissão e recepção de um grupo de comandos escritos em códigos digitais. Nas recomendações ITU, cada comando é designado por iniciais, tipicamente de três letras, como VCF, o qual significa *Video Command Freeze* (comando de congelamento de imagem).

3.4.1. Vídeo, Comunicação e Controle

Em geral, o MCU deve obedecer aos seguintes requisitos:

- Oferecer operação bidirecional ponto-a-ponto com três ou mais terminais;
- Taxa de transmissão de dados igual para os terminais;

- Controle e sinais de indicação;
- Controle de chamada;
- Estrutura de quadros.

3.4.1.1. Comutação de Vídeo (Aparecimento Seletivo)

No modo de troca de vídeo multiponto, cada tela exibida deve corresponder ao vídeo de um terminal. Isto se opõe a mistura de vídeo (*video mixing*), onde o vídeo de mais de um participante é visualizado. Há vários métodos possíveis para selecionar qual vídeo é visto por cada terminal.

3.4.1.2. Comutação por Ativação de Voz

A habilidade do MCU conduzir a conferência usando ativação de voz para determinar qual voz de terminal transmitir a outro é obrigatória. No entanto, esta troca de vídeo pela voz também pode ser feita pelo terminal que preside a sessão.

3.4.1.3. Controle de *Broadcast*

O MCU deve ter habilidade de permitir ao usuário de transmitir em modo *broadcast* para os outros terminais. Deve identificar e cumprir o MCV (*Multipoint Command Visualization-forcing*), bem como cancelar este comando.

O MCV permite a um terminal solicitar ao MCU que seu vídeo seja transmitido em *broadcast* aos outros usuários. O seu cancelamento retorna a conferência para o modo de exibição anterior. A recomendação H.243 descreve isto com detalhes.

3.4.1.4. Controle de Seleção de Usuário

A habilidade do MCU de permitir ao usuário selecionar o vídeo de um terminal qualquer é opcional. Quando esta capacidade é fornecida pelo MCU, este pode identificar e, caso não haja conflito com outros modos, obedece ao VCS e ao Cancel-VCS do usuário terminal.

O VCS (*Video Command Select*) permite ao usuário solicitar que o MCU envie o vídeo ao terminal especificado. Cancel-VCS retorna a conferência ao modo de comutação por ativação de voz.

3.4.1.5. Controle de Cadeira

A habilidade do MCU de presidir o controle da conferência é opcional. Isto é indicado pelo sinal CIC (*Chair-control Indicate Capability*).

Um MCU que possui controle de cadeira pode prover uma conferência com as seguintes habilidades:

- Permitir ao terminal exibir para os outros usuários;

- Permitir ao usuário solicitar o controle da conferência (CCA);
- Permitir ao usuário liberar o controle da conferência (CIS);
- Espalhar o vídeo de um terminal a todos os outros participantes (VCB);
- Retornar a conferência ao comutação por ativação de voz (Cancel-VCB);
- Descartar um terminal da conferência (CCD);
- Encerrar a conferência (CCK).

Um participante da conferência que deseja falar durante uma conferência com um condutor, deve solicitar o direito a este. Uma ação de um participante, como pressionar um botão de requisição no terminal, irá enviar o pedido (TIF) ao MCU. O terminal que preside a sessão indicará que outro solicita este direito. As ações tomadas em resposta a solicitação podem ser as seguintes:

- Ignorar a solicitação;
- Adiar a solicitação enquanto negocia o pedido de outro terminal;
- Trocar o direito de comunicar do terminal por espalhar o vídeo deste para todos os terminais (VCB) e assegurar que o áudio é distribuído aos terminais com mistura de áudio.

A característica seguinte é opcional:

- Solicitar a visualização de um determinado usuário. Em uma conferência com presidência, este comando provê a capacidade de permitir ao presidente de visualizar um terminal enquanto os outros vêem o vídeo previamente selecionado (VCB).

3.4.1.6. FEC Framing on Switching

A capacidade de fazer correção de erro (FEC) é opcional. Quando o terminal que provê o sinal de vídeo é modificado, devido a algum dos procedimentos anteriores, os pacotes de vídeo que são trocados irão causar um atraso na imagem dos terminais, antes que uma nova imagem esteja disponível nestes.

Em transmissões com baixas taxas, isto pode levar menos de um segundo. Este atraso pode ser eliminado se o MCU executa a reorganização de imagens do FEC. Para fazer isto, o MCU deve sempre detectar o erro dos vídeos que chegam e reencaminhar o vídeo com o erro modificado por ele. Este processo ocorre a todo momento, até quando o vídeo não está sendo trocado. Quando a fonte de vídeo é trocada, a correção de erro não será perdida. Se isto é feito, o MCU, deve detectar os erros nos quadros, e inserir um preenchimento nos quadros que saem, a fim de manter a mesma taxa de transmissão.

3.4.1.7. Identificador de terminal

Um MCU pode opcionalmente prover identificação mais precisa dos terminais usando *Terminal ID*. *Terminal ID* permite aos usuários utilizarem sequências alfa numéricas, como nomes ou localizações, bem como números arbitrários. Isto permite aos participantes enxergarem a identificação do provedor do vídeo. O terminal que preside a sessão pode solicitar a identificação dos terminais ao MCU a fim de ajudá-lo a escolher o terminal correto para a transmissão de vídeo. O MCU solicita a identificação aos terminais usando TCI ou TCS. O terminal responde com TII ou IIS. Um terminal pode solicitar a identificação de outro usando TCP. O MCU responde com TIP. TCS e IIS (MBE) são os métodos recomendados.

3.4.1.8. Mistura de Vídeo (aparecimento contínuo)

Mistura de vídeo envolve multiplexação das imagens selecionadas para uma imagem no formato de "tela dividida". Esta é uma característica opcional. Isto requer decodificação e codificação do código de vídeo, e também requer conhecimento da recomendação H.261.

Ainda não foram definidos padrões para combinação de vídeo, por isso, alguns MCU´s dividem em quatro quadrantes, outros em oito quadrantes. Deve-se utilizar este artifício de acordo com os métodos usuais, enquanto não é possível ao terminal prover este controle ao MCU.

3.4.2. Seleção do SCM

O modo de comunicação selecionado (SCM) é um conjunto de taxas de bits, vídeo, áudio e dados, que o MCU tenta manter durante a conferência. Para fazer a comunicação com o MCU, a taxa de transmissão deve ser comum entre todos os terminais, embora diferentes algoritmos de áudio podem sem utilizados. Isto porque o MCU soma os sinais de áudio e os envia em pacotes de mesmo tamanho e mesma taxa a todos os usuários.

O MCU deve determinar o SCM para a conferência. O SCM pode mudar durante a conferência de acordo com os terminais que entram ou deixam a sessão.

É aconselhável que usuário conheça o impacto que o SCM pode ter no andamento da conferência. Por exemplo, se o usuário espera operação com 384 kbit/s usando G.722, ele deve ter certeza que o SCM pode suportar esta capacidade. Os métodos abaixo podem ser utilizados para determinar o SCM:

- SCM é fixado como característica permanente do MCU;
- SCM é determinado automaticamente pelo MCU de acordo com as capacidades dos terminais conectados;

- Muitos SCM's são fornecidos. Um é selecionado pelo MCU no momento que a conferência é estabelecida;
- SCM é determinado usando os procedimentos definidos na MLP (T.120).

3.4.2.1. SCM mínimo

O SCM deve incluir estes modos que possibilitarão a menor interoperabilidade entre os terminais, determinando uma capacidade obrigatória. Pode ser 2.56 kbit/s para áudio, 68.8 kbit/s ou 70.4 kbit/s para vídeo, e 0 kbit/s para dados para alguns usuários, e 2.48 kbit/s par áudio, 60.8 kbit/s ou 62.4 kbit/s para vídeo, e 0 kbit/s para dados para outros usuários.

3.4.2.2. Terminais Secundários

Para determinar a SCM, o MCU deve verificar se muitos terminais possuem capacidades comuns, que são maiores que a dos outros terminais. Os primeiros são chamados de terminais primários, enquanto os segundos são os terminais secundários. Uma capacidade opcional é a de o MCU permitir aos usuários secundários de participar da conferência com funcionalidades limitadas. Por exemplo, um terminal pode participar de uma conferência que todos possuam vídeo e este não. Sem esta capacidade, os terminais secundários seriam eliminados da conferência.

3.4.3. Áudio

O MCU deve possuir capacidade de tratamento de áudio de acordo com a G.711, para lei-A e lei-μ. Isto permite conferência com terminais europeus, que seguem lei-A, e os outros terminais.

3.4.3.1. Mistura de áudio

Mistura de áudio deve ser uma operação padrão do MCU. Deve ser efetuado pela soma dos sinais de áudio recebidos (PCM ou análogo). Em geral, todo sinal de áudio recebido é somado, mas sinais baixos podem ser suprimidos a fim de minimizar interferência em grandes conferências.

Pode-se conectar o áudio de um terminal a outro. Neste caso, os sinais de áudio do terminal não são combinados. Este método é desejável em algumas aplicações como educação a distância, onde sons indesejáveis não são esperados.

Mistura de áudio pode ser usada para conectar terminais em conversações privadas. Seu controle pode seguir os resultados dos comandos de comutação de vídeo, como VCB, ou talvez fora de banda.

Devido ao áudio ser decodificado e codificado, e o vídeo ser comutado, pode haver maior atraso no canal de áudio que no canal de vídeo. Enquanto a compensação de atraso não é requerida,

um atraso no canal de vídeo é permitido para manter sincronização de áudio e vídeo. O atraso de tempo entre os sinais de áudio e vídeo pode ser medido, conforme especifica o anexo C da H.261.

3.4.3.2. Comutação por Ativação de Voz

O MCU pode analisar as entradas de áudio, através de algoritmos, para determinar qual participante será o próximo a tomar a palavra. O resultado deste algoritmo pode ser usado para determinar qual sinal de vídeo transmitir para cada terminal, ou MCU, na ausência de VCB, VCS ou MCV.

3.4.4. Comunicação de Dados

O MCU pode opcionalmente suportar comunicação de dados usando canais de dados de baixa velocidade, canais de dados de alta velocidade, canais de velocidade MLP, e/ou canais de alta velocidade MLP, como definido em H.221. Os canais de dados MLP contêm informação de utilização de protocolos de transmissão para dados multimídia, como definido nas recomendações ITU-T T.120. A série T.120 não inclui protocolos de comunicação de dados e procedimentos, mas inclui aplicações opcionais como transferência de imagem fotografada, anotações, apontamentos, transferência binária de arquivo, e controle da conferência.

A fim de os terminais com capacidade T.120 interagirem com os terminais da conferência multiponto, o MCU pode seguir os procedimentos definidos em H.243 para abertura e fechamento de canais de dados MLP.

3.4.5. Confidencialidade e Segurança

Como uma opção, o MCU pode prover confidencialidade ou operações seguras. No entanto, este aspecto não será abordada nesta discussão devido a extensão do assunto.

3.4.6. Cascateamento

A habilidade do MCU participar de uma conferência que envolva mais de um MCU é opcional e é chamado cascata. Existem dois tipos de cascata: simples e principal/satélite. Quando o número máximo de MCU's a serem conectados é dois, a cascata simples é suficiente. Se são necessários três ou mais MCU's a serem conectados, a cascata principal/satélite é requerida (esta também opera com dois MCU's).

O número máximo de MCU's entre quaisquer dois terminais não pode exceder três. Para uma configuração em estrela, o MCU principal deve ser designado, antes da chamada, como o

MCU no centro da estrela. Em cascata principal/satélite, o MCU principal transmite o comando MIN ao MCU satélite. No caso de disputa para designação do principal, o comando RAN pode ser usado, conforme descrito na ITU-T H.243, procedimento de resolução de disputa. O comando RAN é obrigatório para MCU's que não suportam administração principal/satélite, ou, onde o cliente não deseja fazer uso de administração destas características.

3.4.7. Operação de Conferência Simultânea

Um MCU pode ser utilizado em mais de uma conferência ao mesmo tempo. Isto é conhecido como operação simultânea. O número de conferências simultâneas que podem ser presididas não é padronizado, mas deve ser feito sem prejudicar a qualidade das conferências. Este MCU deve ter requisitos a mais para realizar este tipo de serviço.

3.4.8. Serviços Adicionais Acrescentados

Um MCU pode opcionalmente oferecer serviços adicionais que não estão descritas na recomendação ITU-T H.320.

Alguns destes serviços podem ser ativados pelos terminais que utilizam caracteres SBE. Serviços adicionais acrescentam capacidade adicional a conferência que são acessadas pelo terminal. Estes serviços podem ser código de acesso (senha), solicitação de um operador, acesso ao sistema de reserva, adicionar outro grupo, etc. Estes serviços seriam acessados por sequências de caracteres, como #O. Estas sequências de caracteres não estão padronizadas.

3.4.9. Controle do Terminal de uma Conferência Multiponto

Esta sessão descreve as várias capacidades que o terminal possui em uma conferência multiponto. Três tipos são definidas: capacidade normal de terminal multiponto, capacidade de controle de usuário e capacidade de controle de cadeira. Estas habilidades são especificadas nas recomendações ITU-T H.230 and H.243.

O controle de seleção de usuário é mais eficiente que controle de broadcast. No entanto, controle de cadeira é a técnica de controle multiponto mais eficiente.

3.4.10. Capacidade Normal de Terminal Multiponto

Todos terminais devem possuir capacidade de participar de uma conferência multiponto. Estes terminais possuem as seguintes capacidades em uma conferência multiponto:

- Ver o vídeo enviado pelo MCU;

- Ter seu vídeo espalhado pelo MCU a todos terminais quando este determinar;
- Escutar e ser ouvido pelos terminais;
- Solicitar verbalmente o direito a palavra, em caso de controle de voz ou controle de cadeira;
- Congelar sua imagem durante comutação de vídeo para minimizar corrupção do vídeo (VCF);
- Atualização rápida de seu vídeo para os outros terminais, quando for selecionado como fonte de vídeo pelo MCU;
- Descongelar a tela dos terminais quando realiza atualização rápida, por inserção de quadros congelados no cabeçalho do quadro H.261;
- É recomendável que todos terminais sejam capazes de abrir canais de dados e obedecer MCS e MCN, se não podem processar os dados;
- Todos terminais devem igualar suas taxas de transmissão e recepção ou serem transferidos ao status de terminais secundários (MCC).

Algumas destas características não podem ser suportadas se o terminal é designado como secundário pelo MCU.

3.4.10.1. Capacidades Opcionais

A capacidade opcional seguinte é recomendada ao terminal que possui uma interface de rede mais que um canal físico (como ISDN). Este terminal deve identificar o TIA e transmitir o TIC e TIX. Sem as características abaixo, os terminais não poderão participar de uma conferência multiponto, em determinados tipos de redes:

- Indicação NO AR, quando seu vídeo é enviado aos outros terminais (MIV);
- Indicação quando este é o único terminal conectado a conferência multiponto, mostrando porquê este não possui nenhum áudio ou vídeo se juntam a conferência (MIZ);
- Indicação que este é um terminal secundário na conferência, assim, não pode ter todas vantagens dos outros terminais (MIS);
- Recebe um número terminal do MCU (TIA);
- Solicita uma lista do número terminais de todos participantes da conferência (TCU);
- Obtém e exibe uma lista dos números terminais dos participantes (TIL);
- Obtém e exibe o número terminal de um participante adicionado a conferência (TIN);
- Obtém e exibe o número terminal de um participante que abandona a conferência (TID);

- Obtém e exibe o número da fonte de vídeo (VIN);
- Solicita o direito a controle de cadeira (TIF);
- Responde a solicitação do TCI do MCU por um *Terminal ID* com uma identificação alfa numérica, como nome ou localização do terminal (TII);
- Responde a solicitação do TCS do MCU por um *Terminal ID* com uma identificação alfa numérica, como nome ou localização do terminal (IIS);
- Solicita o *Terminal ID* de outro (TCP);
- Obtém e mostra o *Terminal ID* do outro terminal (TIP);
- Acessa serviços adicionais providos pelo MCU usando caracteres SBE. Podem ser senhas, solicitação de mudanças na configuração da conferência, etc. Estes serviços não estão padronizados, mas requerem que o terminal aceite caracteres SBE.

3.4.10.2. Capacidade de controle de usuário

Terminais com Controle de Usuário possuem todas as características obrigatórias de terminais comuns, e algumas capacidades adicionais que permitem a eles realizarem controle em determinado grau, incluindo o de serem capazes de solicitar que seu vídeo seja espalhado aos outros terminais e receberem as imagens determinado usuário.

3.4.10.2.1. Controle de *Broadcast*

O usuário pode querer que seu vídeo seja espalhado a todos os outros usuários da conferência. Isto é útil, por exemplo, para distribuição de imagem de câmera de documento aos participantes que não comutam o vídeo para o terminal que está falando. Está função é chamada controle de *Broadcast*.

Um terminal que pretenda assumir este controle deve possuir todas capacidades de um terminal comum mais as capacidades adicionais descritas em H.230 e H.243.

Estes comandos provêem o terminal com as seguintes capacidades:

- Solicitar aos terminais verem seu vídeo (MCV);
- Retornar ao modo comutação de vídeo (Cancel-MCV). O Controle de Broadcast é obrigatório em todos MCU's, mas opcional aos terminais. Este controle não é válido em conferência com controle de cadeira.

3.4.10.2.2. Controle de seleção de usuário

O usuário pode controlar o vídeo que os outros participantes da conferência recebem. Esta função é chamada controle de seleção de usuário. Esta capacidade só é efetivada se o MCU suporta este tipo de comando.

Um terminal que queira realizar este tipo de controle deve ter todas características de terminais comuns, mais as capacidades descritas em H.230 e H.243, como o controle usando BAS.

O terminal deve ter meios de obter os números terminais associados aos participantes da conferência. Esta informação é recebida do MCU por VIN, TIN, TID, e TIL.

Estes comandos provêem o controle de seleção com as seguintes capacidades:

- Obtém e exibe os números terminais dos usuários (TCU, TIN, TID, TIL, VIN);
- Solicita ver o vídeo de determinado terminal (VCS);
- Retorna ao modo de comutação de vídeo (Cancel-VCS).

3.4.10.3. Controle de cadeira

3.4.10.3.1. Capacidade básica

Um terminal pode ter a capacidade de executar função de presidente de uma conferência multiponto. Este terminal deve ser capaz de exercer controle sobre a conferência. Esta função é chamada controle de cadeira.

Um terminal que queira realizar este tipo de controle deve ter todas características de terminais comuns, mais as capacidades descritas em H.230 e H.243, como o controle usando BAS.

O terminal deve ter meios de obter os números terminais associados aos participantes da conferência. Esta informação é recebida do MCU por VIN, TIN, TID, e TIL. Assim, o presidente pode comandar o MCU.

Estes comandos provêem o terminal com as seguintes capacidades:

- Obtém e exibe os números terminais dos usuários (TCU, TIN, TID, TIL, VIN);
- Solicita a presidência (CCA);
- Deixa a presidência (CIS);
- Espalha o vídeo de um terminal aos outros (VCB);
- Retorna ao modo voice activated switching (Cancel-VCB);
- Exclui um terminal da conferência (CCD);
- Encerra a conferência (CCK).

3.4.10.3.2. Capacidade Opcional

Capacidades adicionais do controle de cadeira podem ser providas. O controle a seguir pode ser incluído:

- Solicita ver um vídeo específico. Este comando provê uma capacidade de o presidente ver determinados terminais, enquanto os participantes da conferência continuam com a conferência normalmente (VCS).

3.5. Firewall

Firewall [22, 23, 24, 27] é um equipamento ou programa que funciona como proteção de uma rede de dados de acessos não desejados, oriundos de outras redes ou equipamentos. Qualquer equipamento que controle o tráfego por razões de segurança pode ser chamado Firewall.

Em sistemas que utilizam duas conexões de rede, é importante que se possa confiar em todos seus usuários. Pode-se configurar um sistema Linux e atribuir contas a todos que requerem acesso à Internet. Com isto, o único computador da rede privada que conhece tudo sobre a rede externa é o firewall. Assim, qualquer transferência de dados passa pelo firewall e posteriormente para as estações de trabalho.

Apesar de não ser uma ferramenta H.323, o firewall deve atuar no sentido de restringir portas de conexão, permitindo apenas o uso das portas *well known* e aquelas utilizadas durante a sessão de videoconferência. O uso de proxy também é interessante nestes casos, pois permite um maior monitoramento das chamadas.

3.5.1. Políticas do Firewall

Para a utilização do firewall, deve ser definida a política de funcionamento deste. As propostas de um firewall são:

- Manter possíveis invasores afastados;
- Proteger a sua rede privada, implementando políticas de acesso a Internet;
- Não permitir que a rede externa “enxergue” a rede interna.

3.5.1.1. Como criar uma política de segurança

A elaboração da política de segurança é montada baseada nos seguintes aspectos:

- Descrever o que se necessita para o serviço;
- Descrever o grupo de pessoas que se precisa servir;
- Descrever qual serviço de cada grupo;

- Para cada serviço, descrever como este pode ser mantido seguro;
- Relacionar os tipos de violação ao acesso que podem ocorrer.

3.5.2. Tipos de Firewalls

Existem dois tipos de firewalls:

- Firewalls de fitragem – que bloqueiam os pacotes de rede selecionados;
- Servidores Proxy - que fazem as conexões de rede para o usuário.

3.5.2.1. Firewalls de filtragem

Este tipo de firewall foi concebido na plataforma Linux kernel e funcionam no nível de rede. Os dados só são permitidos deixar o sistema caso as regras do firewall permitam. Na medida em que os pacotes chegam, são filtrados por tipo, endereço de origem, de destino, informação de porta de cada pacote, TCP ou UDP, de acordo com seu cabeçalho IP.

Muitos roteadores de rede tem habilidade de executar serviços de firewall. Por isso, torna-se necessário conhecimento de estrutura de pacotes IP para trabalhar com esta função. Devido aos firewalls analizarem e documentarem poucos dados, requerem menos CPU e criam menos latência na rede.

Firewalls de fitragem não fornecem controle por senhas, ou seja, usuários não podem se identificar. A única identificação é o número IP associado a estação de trabalho. Isto pode ser um problema quando se utiliza endereço IP dinâmico, porque as regras são baseadas em números IP e assim, não atinge endereços não cadastrados. Além disso, são mais transparentes ao usuário, pois não têm configurar regras em suas aplicações para usar a Internet. Na maioria dos proxy isto não é verdade.

3.5.2.2. Servidores Proxy

Proxies são muito usados para controle e monitoração de tráfego externo. Existem dois tipos deles:

Existem dois tipos de servidores proxy:

- Proxy de aplicação;
- Proxy de circuito.

3.5.2.2.1. Proxy de aplicação

Quando um cliente quer realizar telnet na rede externa, por exemplo, primeiro é encaminhado ao proxy e então este o conecta ao servidor requerido e retorna os dados ao cliente

novamente. Como o servidor proxy monitora todas as comunicações, pode anotar tudo que é feito. Assim, podem filtrar palavras inapropriadas de sites que são visitados, e também rastrear em busca de vírus.

Também podem autenticar os usuários. Antes de estabelecer uma conexão com a rede externa, podem requerer o *login* inicialmente.

3.5.2.2.2. Proxy de circuito

Um servidor proxy de circuito faz uma ponte entre a conexão do sistema para uma conexão externa, realizando desta maneira uma conexão virtual direta. O mais comum deles é o SOCKS. A maioria dos servidores SOCKS trabalham com conexões TCP, e como firewalls de filtragem, não requerem autenticação, sendo assim, são transparentes ao usuário. No entanto, podem gravar cada conexão dos usuários.

3.5.3. Arquitetura de Firewall

Existem muitas maneiras de estruturar a rede para proteger seus sistemas usando firewall. Se há uma conexão dedicada a Internet através de roteadores, pode-se conectar o roteador diretamente ao firewall. Ou, pode-se passar por um hub para prover aos servidores acesso completo fora do firewall.

3.5.3.1. Arquitetura Dial up

Pode-se utilizar serviço dial up como uma linha ISDN. Neste caso, pode-se usar uma terceira placa de rede para prover filtragem DMZ. Isto permite controle total sobre os serviços de Internet e ainda separa Internet de rede interna, como mostra a figura 3.4 a seguir:

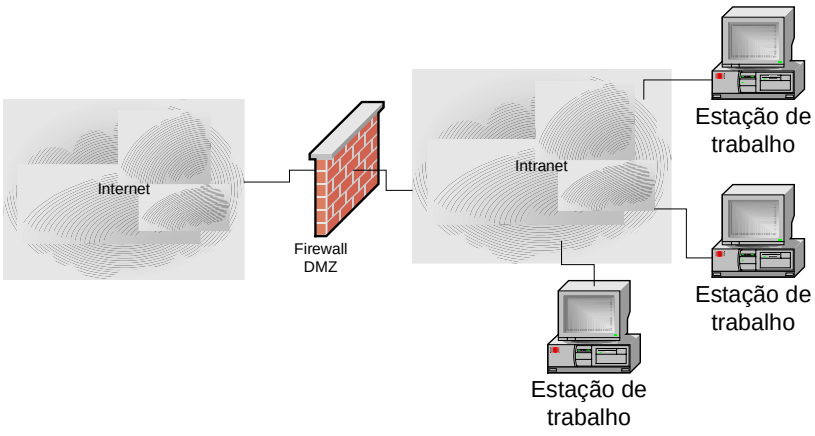

Figura 3.4 – Esquema da arquitetura dial up [22]

3.5.3.2. Arquitetura de roteador simples

Em casos que haja roteador entre o firewall e a Internet. Quando tem-se controle do roteador, pode-se configurar regras de filtragem severas nele. Caso contrário, não se pode fazer este tipo de controle, a menos que se faça acordo com o provedor de acesso a Internet, como mostra a figura 3.5:

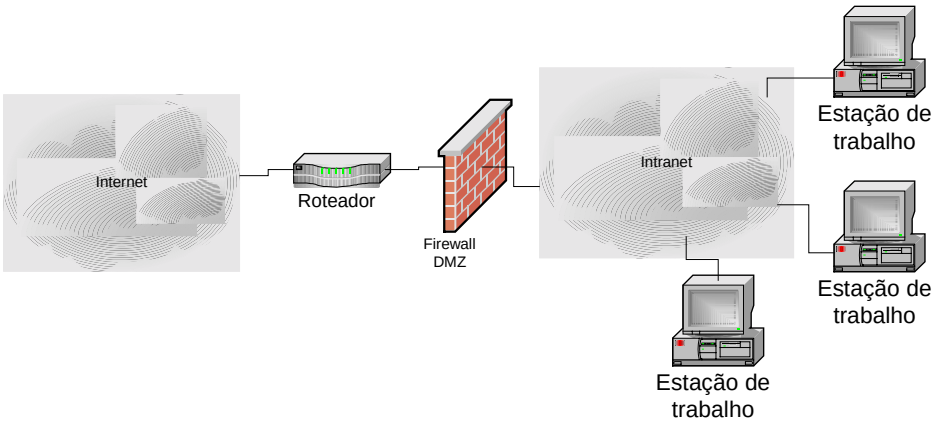

Figura 3.5 – Esquema da arquitetura de roteador simples [22]

3.5.3.3. Firewall com Servidor Proxy

Para monitorar onde os usuários da rede estão indo e a rede é pequena, pode-se integrar um servidor proxy ao firewall. Servidores de acesso à Internet fazem isto algumas vezes para revender a agências comerciais, como indica a figura 3.6.

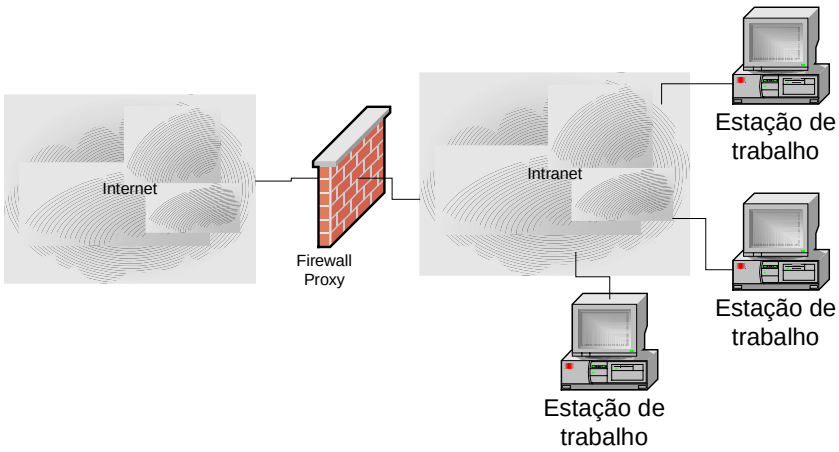

Figura 3.6 – Esquema da arquitetura firewall com servidor proxy [22]

Pode-se colocar o servidor proxy na LAN. Neste caso, o firewall deve ter regras para permitir somente acesso do proxy a Internet. Deste modo, os usuários só podem acessar a Internet através do proxy. A figura 3.7 ilustra a situação.

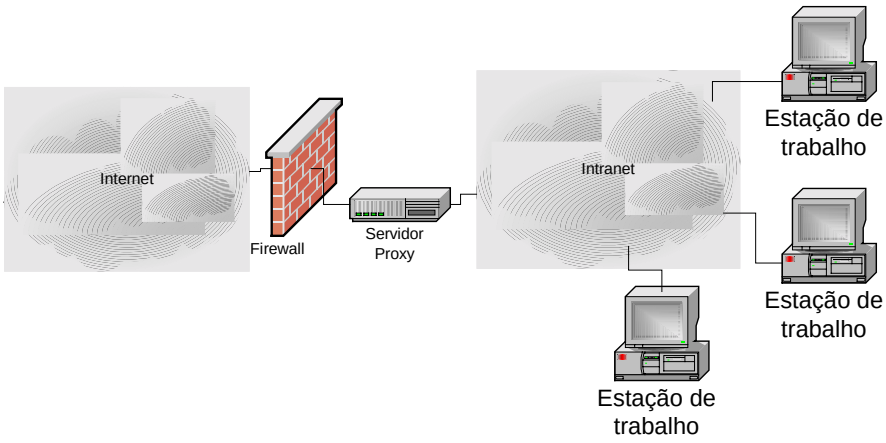

Figura 3.7 – Esquema da arquitetura servidor proxy na LAN [22]

3.5.4. Segurança do Firewall

O firewall não é um sistema robusto a ataques, apesar de seguro. Para minimizar as possibilidades de invasão, deve-se restringir ao máximo o seu acesso. No arquivo /etc/inetd.conf pode-se restringir o acesso das portas *well known*. Desabilitando, por exemplo, echo, discard, daytime, chargen, ftp, gopher, shell, login, exec, talk, ntalk, pop-2, pop-3, netstat, systat, tftp, bootp, finger, cfinger, time, swat and linuxconfig.

Devido ao firewall apenas verificar o pacote no momento que chega, pacotes podem ser criados simulando outras máquinas, caracterizando assim uma forma de ataque.

Não garantem proteção contra ameaças internas. Um firewalls é um ponto de entrada para uma rede, portanto não pode oferecer proteção contra um tráfego que não passa por ele. Exemplo - ataques praticados dentro da própria organização (ataques interno) e os dados que são transmitidos e recebidos através de modems de discagens, das mesas dos funcionários. Um perímetro de rede seguro implica que todos os pontos de entrada da rede sejam seguros.

Para se definir as regras do firewall, utilizam-se softwares específicos habilitados para tal. Neste trabalho, utilizou-se o Ipchains.

3.5.5. Fundamentos dos Filtros de Pacotes

3.5.5.1. O que são filtros de pacotes

Todo tráfego através de uma rede é feito na forma de pacotes. Os pacotes possuem duas divisões básicas: o cabeçalho e o corpo.

Um filtro de pacote é um *software* que analisa o cabeçalho dos pacotes e decidem o destino de cada um deles. Podem descartá-lo, encaminhar o pacote, ou rejeitá-lo, neste caso, além de

descartar, encaminha mensagem a origem comunicando que isto foi feito. Os termos utilizados pelos filtros para cada uma destas atitudes são: *deny, accept e reject.*

Estas regras são usadas para se ter bom controle, segurança e cuidado. Controle para permitir somente determinados tipos de tráfego; segurança para proteger ao máximo a rede interna de ataques; cuidado ao deparar com situações anormais de troca de pacotes.

O IPChains é uma ferramenta que permite inserir ou excluir regras aos filtros de pacotes.

3.5.5.2. Como os pacotes atravessam o filtro

O kernel inicializa com uma lista de três regras, chamadas chains. Chain é uma lista de regras. Os chains são *input, output* e *forward*. Quando um pacote chega, o kernel analisa a regra de *input* para decidir seu destino. Se ele é aceito, o kernel decide para onde enviar este pacote, processo este chamado de roteamento. Se seu destino é outra máquina, será consultado a regra de *forward*. Se este está habilitado, o kernel consulta finalmente a regra de *output*.

Cada regra diz respeito ao cabeçalho do pacote IP e o que fazer com ele. Se a regra não comtempla o pacote, então a próxima regra da cadeia é consultada. Quando não há mais regras a serem consultadas, o kernel verifica a política do chain para decidir o que fazer. A política do kernel normalmente orienta o kernel a rejeitar ou descartar o pacote.

O esquema do ipchains está mostrado na figura 3.8 a seguir.

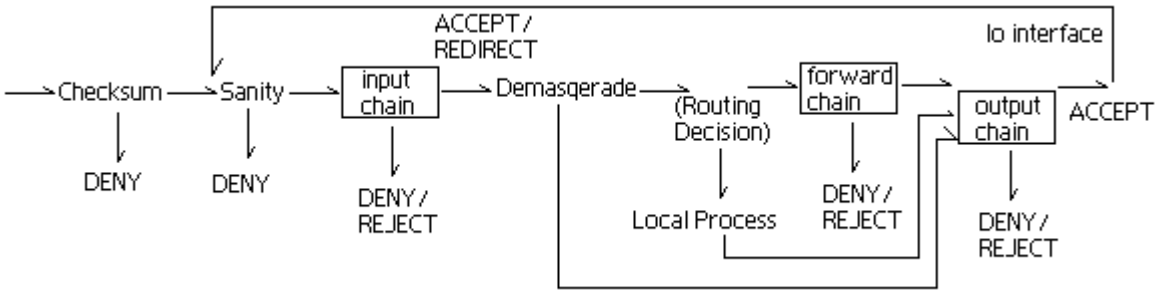

Figura 3.8 – Esquema do ipchains [24]

Passo a passo, as etapas são as seguintes:

- Checksum: Teste para verificar se o pacote não foi corrompido em alguma etapa. Em caso positivo, é negado;
- Sanity: Alguns pacotes mal formados podem confundir o código de checagem de regras, por isto são negados neste ponto e uma mensagem é gerada no syslog;
- Input chain: Primeira regra do firewall contra qual pacote será testado. Se a decisão não for negá-lo, então o pacote segue;

- Demasquerade: Se o pacote é uma resposta a um pacote previamente mascarado, é desmascarado e encaminhado a regra de output. Se o pacote não está mascarado, pula-se esta etapa do diagrama;
- Routing decision: O campo de destino é examinado pelo código de roteamento para decidir se o pacote será processado localmente ou encaminhado ao terminal;
- Local process: Um processamento pode ser feito após a decisão de roteamento, e então encaminhar os pacotes adiante;
- lo interface: Se os pacotes do processador local são para outro deste tipo, irão para a regra de output com a interface setada com Io;
- local: Se o pacote não foi gerado pelo processador local, a regra forward é verificada; caso contrário vai para a regra output;
- forward chain: Esta regra é aplicada a todos pacotes que tentam passar desta para outra máquina;
- output chain: Esta regra é aplicada por todos os pacotes antes que eles sejam enviados.

3.5.6. IP Masquerade

IP Masquerede é um forma de translação de endereço de rede (NAT) que permite que computadores conectados em uma rede interna não tenham um ou mais registro de endereço IP, habilitados para comunicar com a Internet.

IP Masquerade é uma função do Linux similar ao NAT (*Network Address Translator*), presente em firewalls comerciais e roteadores de rede. Por exemplo, se o *host* Linux está conectado a Internet via PPP, Ethernet, etc, o IP Masquerade permite que outro computador da rede interna, operando com Linux, também se conecte a Internet. Assim, o IP Masquerade permite esta funcionalidade mesmo que as máquinas da rede interna não possuam endereço IP fixo.

MASQ permite a um grupo de computadores de serem “invisíveis” à Internet. Para os computadores que estão na Internet, todo tráfego de dados vindos da rede interna aparecerá como sendo advindo do servidor Linux da rede. Assim, o IP Masquerade permite a criação de um ambiente de rede seguro. Com um firewall bem configurado, quebrar a segurança de um sistema mascarado de rede interna é considerado difícil.

3.5.6.1. Status

IP Masquerade vem sendo usado durante anos em Linux com kernel estágio a partir de 2.2.x. Kernels desde o Linux 1.3.x tem o MASQ associado a sua concepção. Hoje, muitas

aplicações individuais e de negócios, que o utilizam, vêm obtendo excelentes resultados. Redes comuns que utilizam browsers Web, com TELNET, FTP, PING, TRACEROUTE, etc, operam bem com o IP Masquerade.

IP Masquerade opera bem em servidores de clientes que possuem diferentes plataformas. Estas podem ser:

- Unix: Sun Solaris, *BSD, Linux, Digital UNIX, etc;
- Microsoft Windows 2000, NT (3.x and 4.x), 95/98/ME, Windows para Workgroups (com pacotes TCP/IP);
- IBM OS/2;
- Apple Macintosh MacOS operando com MacTCP ou Open Transport;
- Sistemas baseados em DOS com dispositivos de pacotes e pacotes Telnet NCSA;
- VAXen;
- Compaq/Digital Alpha operando em Linux e NT.

3.5.6.2. Beneficiados com o uso do IP Masquerade

Usuários que possuam um *host* Linux conectado à Internet e possui computadores utilizando TCP/IP conectados em rede local, ou um *host* Linux que possui um ou mais modems, utilizando servidor PPP ou SLIP, conectando os outros computadores, e estes não possuam endereço IP oficial. Então pode-se comunicar com a Internet sem precisar de endereços IP oficiais, o que representa economia de dinheiro, pois gastos com novos endereços IP não serão necessários, além de garantir certa segurança.

4. Áudio e Vídeo *on-demand* (*Streaming*)

4.1. *Definição de* Streaming

Streaming em informática é o processo de enviar em pacotes (*streams*) um arquivo e, consequentemente, lê-lo enquanto ainda está se fazendo o seu *download*.

Streaming de Vídeo é uma tecnologia de transmissão de vídeo através de redes de comunicação (como redes locais e/ou Internet), que possibilita aos usuários visualizarem o conteúdo de um vídeo sem a necessidade de fazer o seu *download* para as suas máquinas. O vídeo é dividido em sessões, o que permite que seja visto à medida que chegam. Pode-se dizer que o *streaming de video* está para o vídeo "tradicional" como o *browser Netscape* está para o *Mosaic*, na apresentação de páginas *web*. No primeiro as páginas *web* são exibidas enquanto são carregadas, ao passo que no segundo, isso só acontece quando o conteúdo estivesse todo "carregado".

4.2. Vantagens e desvantagens relativamente ao vídeo "tradicional"

A grande vantagem de usar *streaming de vídeo* é a possibilidade de visualizar o vídeo quase (breve em tempo real) no instante em que se deseja vê-lo. Em vídeo tradicional tem-se que primeiro fazer ou seu *download*. Neste aspecto, o primeiro bem mais atraente que o segundo. Até permite se ver um pedaço do vídeo e depois cancelá-lo, caso este não interessar ao usuário.

A grande desvantagem do *streaming de vídeo* relativamente ao vídeo "tradicional" é o fato de, regra geral, estar se impossibilitando o armazenamento dos pacotes em disco para o caso de utilização futura, por exemplo, e também necessita de recursos, como largura de banda, para uma ótima visualização.

4.3. Potencialidades

As grandes potencialidades do *streaming de vídeo* encontram-se ao nível das telecomunicações, do comércio *on-line*, entretenimento e da educação.

Em telecomunicações, é possível a transmissão de som e imagem em tempo real, através da Internet, possibilitando a videoconferência, sem a necessidade de recursos adicionais.

No comércio on-line, é possível, por exemplo, ter uma melhor ideia do que irá se comprar, pois é possível "vê-lo" tal como é – pelo menos assim é esperado.

Em nível de entretenimento, por exemplo, é possível ver *trailers* de filmes, permitindo assim fazer uma melhor seleção do que se assistirá no cinema. (E com os novos adventos, como a Internet 2, será possível ver filmes *on-line* com qualidade TV ou mesmo superior). Na educação é possível o ensino à distância através da Internet já com uma qualidade bastante aceitável e com a possibilidade de *feedback* em tempo útil por parte de alunos e professores.

4.4. *Áreas onde atualmente se aplica o* Streaming de Vídeo

Atualmente existem já áreas bastante diversas onde se aplica o *streaming de vídeo*, desde o vídeo integrado em páginas *web*, para fins lucrativos, até o vigilância por vídeo à distância.

A seguir, é apresentada uma lista, com uma breve descrição, de aplicações comuns do *streaming de vídeo:*

- Multimídia para a *Internet*: como o *RealVideo* ®, Quick Time® e o Media Player®;
- Comunicações pessoais: aplicações como a videoconferência, entre duas ou mais pessoas;
- Correio eletrônico multimídia: envio de correio eletrônico com vídeo em anexo;
- Vigilância por vídeo: segurança a longa distância com vídeo através de uma rede;
- Sistemas de emergência: aplicações de vídeo remotas tais como procedimentos médicos, policiais, militares, etc;
- Difusão de informação: difusão de informação a uma grande quantidade de pessoas;
- Serviços de bases de dados e arquivos: visualização vídeos de uma biblioteca ou arquivo da mesma maneira que se pode requisitar um livro de uma biblioteca;
- Jogos de vídeo interativos: jogos que usam vídeo, onde o usuário pode escolher o fim de um número de cenários possíveis.

4.5. Princípios de Operação

A Internet utiliza alguns protocolos: o TCP, o IP e também o UDP. O protocolo TCP (*Transmission Control Protocol*) é usado para trocar informação através da rede. Este é um protocolo seguro que garante a correcção de dados (detecção de erros) e por este mesmo fato é mais lento que outros protocolos. O UDP (*User Datagram Protocol*), não faz correção de dados, e por

isto torna-se mais leve. Ambos funcionam sobre o protocolo IP (*Internet Protocol*). A maioria dos programas de *streaming de video* utilizam o UDP, optando por rapidez de transmissão à sua correção de erros. Isto porque é mais aceitável ver – ou não – um *frame* falhado, a esperar grandes porções de tempo, o que provoca impaciência no usuário. *O streaming de video* pode ser transmitido através de HTTP (*HyperText Transfer Protocol*), sendo esta variante mais lenta que as outras, ou através de protocolos que surgiram (alguns ainda em evolução no seio no IETF, e não só) para dar suporte à tecnologia de *streaming*, apresentadas a seguir:

- **RSVP** – *Resource Reservation Protocol:* este é o padrão para reservar largura de banda para que os dados cheguem ao seu destino precisa e rapidamente;
- S**MRP** – Simple Multicast Routing Protocol: é um protocolo que suporta conferência multiplicando os dados para um grupo seleto de receptores, tal como no IP *Multicast.*
- **RTSP** – Real-Time Streaming Protocol: usado especificamente para fazer o *streaming* de programas multimídia para controle de dados em tempo real. Este protocolo já é utilizado atualmente por alguns *routers* e suportado por sistemas de *streaming de multimídia*. Este é um padrão adotado internacionalmente.
- **RTCP** – Real-Time Control Protocol: protocolo de controle da qualidade de serviço (QoS – *Quality of Service*), para garantia de qualidade de ponto-a-ponto.

Tanto estes protocolos, como o HTTP são usados sobre UDP ou TCP. A hierarquia de protocolos é mostrada na tabela 4.1.

Tabela 4.1 – Hierarquia dos protocolos (para *streaming*)

Protocolo	**Camada**
HTTP, RTSP	Nível de Aplicação
TCP, UDP	Nível de Transporte
IP	Nível de Rede
...	Níveis Enlace de Dados e Físico

Para a transmissão da informação através da rede, secciona-se os vídeos em pequenos pedaços de tamanho reduzido e envia-os. Estes pacotes devem ser passíveis de serem lidos por uma aplicação ou *plug-in* adequado, de maneira a serem automaticamente reproduzidos logo que chegam ao seu destino. Para se fazer o *streaming* de um vídeo, apenas é necessário um codificador que

converta vídeo nos formatos comuns utilizados (AVI, MPEG, MOV) para um dos formatos proprietários de *streaming* existentes no mercado, uma vez que ainda não existe um formato padrão. Este codificador não comprime nem altera as propriedades de um vídeo.

As extensões comuns de *streaming de vídeo* são mostradas na tabela abaixo:

Tabela 4.2 – extensões de *streaming* comuns

Extensão	**Descrição**
.asf	Advanced Streaming Format (Microsoft ®)
.rm	Real Video/Audio File (RealNetworks ®)
.swf	Shockwave Flash (Macromedia ®)
.viv	Vivo Movie File (Vivo Software ®)

4.6. Implementação

Para implementar o *streaming de vídeo* são necessários *hardware* e *software* para o servidor de *streaming*, e outro conjunto deste para o usuário.

Em nível do *hardware*, é preciso um servidor, que pode ser um servidor *web*, mas que suporte alta largura de banda e acesso a muitos usuários; para otimizar o processo de *streaming*, é recomendado um servidor dedicado ao *streaming*, e outro como servidor *web* (HTML, imagens, etc). Caso o *streaming de vídeo* enviado para a rede seja em tempo real, é aconselhável a utilização de um computador exclusivo para a captura e posterior digitalização/compressão do vídeo. Todos estes computadores devem ser ligados ao servidor com cabos que permitam alto débito.

Para o usuário, é necessário considerar a largura de banda e modem que este dispõe; não se pode fornecer um arquivo de *streaming* otimizado para modems de 56 kbps a usuários com modems de 28.8 kbps, por exemplo.

A nível de *software*, para *streaming de vídeo*, há várias opções disponíveis das quais duas serão apresentadas nas tabelas 4.3 e 4.4, a seguir, para o Real Video e para o Windows Media Player.

Tabela 4.3 – Prestação de vídeo do Real Video

Item	**Características Vídeo e Requisitos**		
Tipo de Ligação	28.8k Modem	56k Modems	ISDN, T-1, ADSL Cable Modem, Satellite Dish
Velocidade de recepção de vídeo requerida (kbps)	21	37	80
Qualidade vídeo e áudio	Boa	Muito boa	Muito boa
Tamanho do vídeo de amostra	37 s	37 s	37 s
Tamanho do ficheiro (kBytes)	104	179	379

Tabela 4.4 - Prestação de vídeo do Windows Media Player

Item	**Características Vídeo e Requisitos**		
Tipo de Ligação	28.8k Modem	56k Modems	ISDN, T-1, ADSL Cable Modem, Satellite Dish
Velocidade de recepção de vídeo requerida (kbps)	21	37	80
Qualidade vídeo e áudio	Boa	Muito boa	Ótima
Tamanho do vídeo de amostra	37 s	37 s	37 s
Tamanho do ficheiro (kBytes)	107	172	388

4.6.1. Componentes

Para implementar o *streaming de vídeo* em páginas *web*, há sempre três componentes: cliente, servidor e codificador.

O codificador serve essencialmente para converter arquivos de vídeo no formato AVI, MPEG e outros, para o formato do *streaming* utilizado.

O cliente são as aplicações, ou *plug-ins*, que os usuários têm nos seus computadores e que lhes permitem visualisar o *streaming*. Nem todas as aplicações utilizam o cliente.

Finalmente, os servidores contêm o *software/hardware* necessários para fornecer o *streaming* às páginas *web*. Algumas arquiteturas, utilizam s*erver-less*, ou seja, sem servidores.

4.7. Tipos de Arquiteturas

As três arquiteturas de *streaming* que existem são:

- **Cliente/Servidor**: é talvez a mais comum, sendo necessário um codificador para converter os vídeos, um servidor para os disponibilizar e finalmente um cliente para lê-los. Ex: *RealPlayer* ® e *Windows Media Player* ®.
- **Server-less** (sem servidor): utiliza-se o protocolo HTTP para disponibilizar o *streaming*, não usando um servidor dedicado para *streaming*. É, no entanto, mais lenta que as outras duas.
- **Client-less** (sem cliente): é semelhante à arquitetura Cliente/Servidor, onde não é necessário *plug-ins* ou aplicações para lerem os pacotes. Utiliza *Java Applets* para ler os ficheiros, linguagem comum à quase totalidade dos *browsers* – os chamados *java-enabled browsers*. É esta, segundo os especialistas de *streaming de vídeo* e *áudio*, a arquitetura com um futuro mais promissor.

4.8. IP Multicasting

O *streaming de vídeo* consome muitos recursos, sobretudo no que diz respeito a largura de banda. Por isso, é necessário procurar soluções para este problema. Uma maneira é aumentar a largura de banda e comprar equipamentos com maior capacidade. A outra é utilizar *multicasting* através do IP. Com esta tecnologia são obtidos ganhos significativos no que diz respeito à sobrecarga da rede. Esta técnica consiste em mandar só um pacote de informação para todas as máquinas ao invés de mandar uma cópia para cada uma delas, aliviando o tráfego da rede de maneira considerável.

4.9. Video Codecs

Codec é o algoritmo que permite codificar/decodificar métodos de compressão de áudio/vídeo. Alguns dos mais utilizados são os do **Real Video**, da RealNetworks, o **Windows Media Encoder** da Microsoft, e o **QuickTime Encoder**.

4.9.1. Largura de Banda

O streaming de vídeo, baseia-se sobretudo na gestão da largura de banda disponibilizada.

Pouca largura de banda para a disponibilização dos conteúdos multimídia implica pouca performance e um grau elevado de descontentamento por parte dos usuários. Porém, um excesso desta, implica no fornecimento de conteúdos com maior qualidade.

Existem três aspectos fundamentais para a gestão da largura de banda:

- Largura de banda mínima: largura de banda "do cliente" (*bandwidth target*);
- Largura de banda total necessária: por parte do servidor;
- Fatores que influenciam a largura de banda.

4.9.1.1. Largura de banda mínima

Refere-se a dimensão do vídeo que se quer transmitir; assim determina-se a largura de banda mínima necessária para a sua visualização.

No caso de transmissões que ocorre muita movimentação e que possui um campo de filmagem amplo, são pouco aceitáveis transmissões em baixas taxas. Isto porque a janela de vídeo seria de tamanho reduzido, como 160×112 ou 176×144, e seria o mesmo que ver formigas correndo de um lado para o outro.

Por isto é necessário saber qual o público alvo, e fornecer-lhes conteúdos suportáveis pela largura de banda máxima a que estes têm acesso.

4.9.1.2. Largura de banda total necessária

A fórmula para calcular a largura de banda necessária é a seguinte:

Largura de banda total necessária = n.º de ligações × largura de banda (4.1)

Exemplo:

25 pessoas a 28.8 Kbps e outras 25 a 56 Kbps conectadas simultaneamente

largura de banda total = (25 * 28.800 bps) + (25 * 56.000 bps)

= 720.000 bps + 1.400.000 bps = 2.120.000 bps = 2.12 Mbps

Depois de estimar a largura de banda, é necessário colocar linhas físicas que a suportem

4.9.1.3. Fatores que influenciam a largura de banda

- Tamanho dos frames
- Frame rate

- Qualidade de imagem
- Número de cores

Exemplo:

Um vídeo com uma resolução de 160×120 pixels com um *frame rate* de 10 *frames*/segundo e cada *pixel* representado por 24 bits de cor, tem-se:

{[(160×120) ×10] ×24} = 4608000 bps = 4608 kbps

4.9.2. Outros

Outro problema existente, comum a outras tecnologias, é a falta de um formato universal para o *streaming de vídeo* que evite a multiplicidade de aplicações e *plug-ins* necessários para visualizar vários conteúdos de diferentes páginas. Este problema ser solucionado em parte pelas aplicações/arquiteturas c*lient-less*, pois não necessitam de aplicações ou *plug-ins* para lerem os referidos *streaming*.

4.10. Futuro do Streaming *de Vídeo*

Com o rápido crescimento da tecnologia, surgimento de novas tecnologias, novas plataformas mais rápidas, mais eficientes, com conceitos revolucionários, o futuro da tecnologia de *streaming*, e em particular da tecnologia de *streaming de vídeo*, é incerto.

As tecnologias estão sendo desenvolvidas para permitir uma maior rapidez na transferência de dados, tanto a nível de redes locais como mundiais e, claro, da Internet. Em redes locais, existe a tecnologia 1 Gigabit, outras mais baixas como a de 100 Mbits, ou mesmo a de 10 Mbits.

Na *Internet*, tecnologias como ADSL e as suas variantes, já conseguem ver vídeos em tempo real, e com elevada qualidade tanto de imagem como de som. Futuramente, com o aparecimento da Internet 2, essas velocidades tendem a aumentar, tornando o *streaming de vídeo* e todos os restantes materiais multimídia mais atrativos.

Com toda a investigação e desenvolvimento (I&D) tanto na área de multimídia, como na linhas físicas, estima-se futuramente um vídeo em *full screen* e em tempo real. Contudo nada disto será possível se os ISP não disponibilizarem maior largura de banda, se não existir um sistema tarifário único, reduzido, de acesso à *Internet*, e disponibilização de linhas físicas mais rápidas que permitam transferências superiores a 56 kbps. O *Intelligent Streaming* começa a aparecer, e consiste

numa tecnologia que detecta o nivel de tráfego na rede e com isso adapta as propriedades dos *streaming* de vídeo para maximizar a qualidade.

Embora a qualidade de imagem do *streaming de vídeo* ainda esteja longe da qualidade TV, VHS e ainda para difusão em grande escala, o usuário comum pode emitir vídeo através da Internet para qualquer localidade de uma maneira bastante fácil e acessível. Isto dá ao *streaming de vídeo* um papel importante no meio das comunicações.

Não se pode esquecer que esta tecnologia ainda é bastante recente e em breve, com a competição e constante investigação nesta área, poderá assumir um papel de grande relevo no futuro da Internet e consequentemente em tudo o que é relativo a ela.

5. Ambiente de Colaboração Visual

O ambiente proposto para a implementação de um sistema de videoconferência multiponto está descrito na figura 5.1 abaixo:

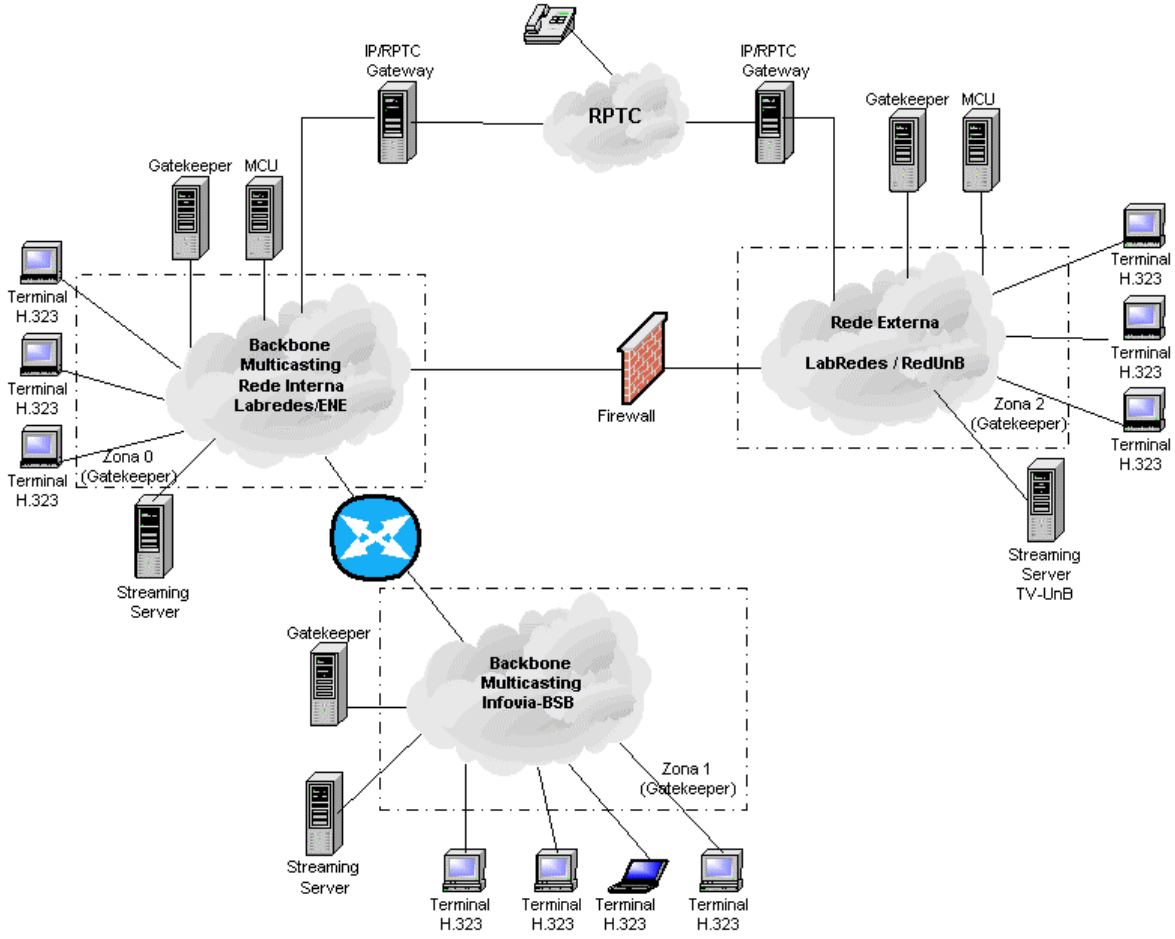

Figura 5.1 – Ambiente de videoconferência proposto

A figura mostra basicamente três zonas distintas, cada qual sendo controlada por um gatekeeper correspondente. Conforme descrito no capítulo 3, uma das funções do gatekeeper é o gerenciamento de zonas, além do controle de acesso e banda dos terminais registrados a ele. Cada zona está relacionada a um backbone, sendo um deles o da rede Infovia de Brasília, RMAV de Internet2, implementada usando tecnologia ATM e desenvolvida por um consórcio entre UnB, Codeplan, Embrapa e Brasil Telecom (Telebrasília), sendo suportada pelo CNPq e pelo ProTeM/CC.

Os *streaming server* possuem a função de fazer *streaming* de vídeo, possibilitando aos terminais assistirem a filmes ou apresentações gravadas, por exemplo. Esta entidade é bastante

interessante, pois opera também em redes multicast, e em um ambiente como este, com taxas de transmissão relativamente altas, torna-se uma opção a mais de conforto para o usuário. O MCU também está presente no ambiente, pois propicia a comunicação entre mais de dois terminais. Este dispositivo deve ter boa capacidade de processamento para que não haja atraso na recepção e reenvio dos pacotes de áudio, vídeo e/ou dados, o que acaba prejudicando a interatividade da comunicação. Entre os backbone que simulam a rede interna e externa, é proposto a utilização de um firewall. Este é um equipamento ou programa que funciona como proteção de uma rede contra acessos não desejados, oriundos de outras redes ou equipamentos. Pode realizar também o mascaramento de endereços da rede. No caso acima, pretende-se com a utilização deste, oferecer certa segurança ao backbone interno, além de verificar o mascaramento dos pacotes que saem deste, rumo ao backbone externo.

É proposto ainda a interconexão com a rede de telefonia pública comutada, por meio de gateway, obtendo assim um ambiente de comunicação de áudio e vídeo completo. Este tipo de implementação favorece o uso de voz sobre IP (VoIP), assunto também bastante pesquisado na atualidade.

Para a implementação do backbone proposto, é proposto alguns tipos de testes com os componentes utilizados. Estes constituem no estabelecimento de diversas formas de comunicação entre os clientes H.323, desde conexão simples ponto-a-ponto, até a montagem do ambiente completo.

- **Conexão Cliente – Cliente sem Gatekeeper**: Verificação da forma com que os dois clientes se comunicam, desde a chamada inicial, até o estabelecimento da sessão. O acompanhamento das chamadas pode realizado com um software de análise de protocolos, onde se pode observar os pacotes que transitam entre as duas estações;
- **Conexão Cliente – Cliente com Gatekeeper**: Verificação da forma como as duas estações se registram ao Gatekeeper, desde o pedido de conexão de uma das estações até o início da sessão. Pode-se monitorar a chamada do Gatekeeper e se a sessão não apresenta problemas quando o pedido de conexão era alternado entre os dois terminais.
- **Conexão Cliente/Gatekeeper – Cliente/Gatekeeper**: Verificação da maneira como os terminais se comunicavam, por meio de dois Gatekeeper. Pode-se monitorar a requisição de chamada proveniente do cliente A, o registro no Gatekeeper A, a comunicação entre estes dois Gatekeeper, a comunicação do Gatekeeper B com o cliente B, e finalmente o estabelecimento da sessão.

- **Conexão Clente – Cliente – Cliente com Gatekeeper e MCU:** Verificação da interoperabilidade entre três terminais por meio de um MCU e um Gatekeeper. Controlou-se o pedido de conexão proveniente de cada um dos terminais, a comunicação deles com o MCU, e a troca de informação entre este e o Gatekeeper. Foi monitorada qualidade de áudio e vídeo, bem como a distinção dos deles para cada usuário.
- **Conexão Cliente/MCU – Cliente/MCU:** Verificação da interoperabilidade entre dois MCU's se comunicando e comportamento com relação a distribuição de áudio e vídeo.
- **Conexão Cliente – Firewall – Cliente:** Verificação da comunicação entre os terminais trocando pacotes por meio de Firewall. Pode-se fazer restrições de acesso ao Firewall para que um dos clientes mantenha certa segurança. Neste caso, este cliente representa uma rede interna, e outro uma rede externa.
- **Conexão Cliente – Firewall – Cliente com Gatekeeper:** Verificação do estabelecimento da chamada nas condições acima citadas, com um Gatekeeper ligado a rede. Pode-se monitorar a qualidade do serviço obtida.
- **Conexão Cliente/Gatekeeper – Firewall – Cliente/Gatekeeper:** Verificação do estabelecimento da sessão, quando duas redes distintas, a primeira simulando uma rede interna, a segunda simulando uma rede externa, se comunicam através do Firewall.
- **Conexão Cliente/Gatekeeper/MCU – Firewall – Cliente/Gatekeeper:** Da mesma maneira anterior, deve-se verificar o estabelecimento da sessão e o sucesso da conferência multiponto.
- **Conexão Cliente/Gatekeeper/MCU – Firewall – Cliente/Gatekeeper/MCU:** Verificação de dois MCU's se comunicando através do firewall.
- **Conexão Cliente (H.323) – Gateway – Cliente (Telefone):** Verificação da comunicação com o Gateway, quando a chamada é originária do cliente H.323. Pode-se fazer o acompanhamento dos pacotes trocados entre estes, bem como a comunicação entre o Gateway e o cliente – telefone. Neste teste, o telefone pode ser simulado com um modem.
- **Conexão Cliente (H.323)/Gatekeeper/MCU – Gateway – Cliente (Telefone):** Verificação da comunicação com o gateway quando estão envolvidos gatekeeper e MCU.

6. Implementação do Ambiente

6.1. Bibliotecas OpenH.323

O estudo deste trabalho foi baseado em bibliotecas de código aberto. Estas bibliotecas podem ser adquiridas por qualquer pessoa sem custos para isso, e podem ter seu código alterado, já que esta é a filosofia deste tipo de ferramenta. Para o usuário, isto representa uma enorme vantagem em relação aos softwares de código fechado, pois não podem controlar seus parâmetros para otimização do sistema.

O OpenH323, desenvolvido pelo grupo australiano Equivalence Pty Ltd [31], é um projeto bastante interessante na área de videoconferência, no qual todas entidades H.323 são implementadas por meio de software de código aberto. Os códigos fonte são todos licenciados publicamente pelo MPL (Mozilla Public License)[8]. Na *homepage* do grupo são encontrados os arquivos contendo os código fonte para as mais diversas implementações H.323 que se queira realizar, como MCU ou gateway. As bibliotecas para serem compiladas, ou simplesmente os seus arquivos executáveis estão disponíveis na sessão *download*, com a opção de Windows ou Linux, e são constituídas basicamente por PWLib, OpenH323, OhPhone, OpenPhone, OpenMCU, OpenAM e PSTNGW.

O PWLib e o OpenH323 são as bibliotecas básicas para compilação e consequentemente, execução, de qualquer outra outra das bibliotecas citadas. Os procedimentos para compilação destas, e outras bibliotecas utilizadas neste estudo constam no anexo II. A biblioteca OpenH323 possui os códigos fonte para execução das mais diversas funções descritas na própria Recomendação H.323, como codecs de áudio, canais lógicos ou *jitter*. Cada um destes itens corresponde a um arquivo diferente, sendo assim, pode-se modificar o código de um ou outro arquivo, sem muita dificuldade e testar o sistema para verificar o êxito da alteração. Ainda é possível fazer que esta alteração seja de conhecimento comum, ou mesmo do uso de todos, bastando enviar para a lista de discussão da *homepage* o que foi feito. Se for realmente interessante para o aprimoramento do sistema, esta alteração é incluída nas bibliotecas e será distribuído a partir de então.

6.2. Aplicações do OpenH323

6.2.1. OhPhone – terminal H.323

Outra biblioteca interessante é o OhPhone, que realiza as funções do terminal H.323. O OhPhone possui as funções de um terminal como o NetMeeting, podendo enviar/receber ou não itens como áudio, mas não é uma interface gráfica, não executando funcionalidades como o vídeo. Para isto, pode-se utilizar outros software, denominados *front end*, como o gong, utilizado neste trabalho e disponibilizado na sessão *CVS Repository*, também na *homepage* do grupo. Esta sessão será comentada no item 6.2.3.

6.2.2. OpenMCU

O próximo produto utilizado no trabalho e desenvolvido pelo grupo OpenH323 é o OpenMCU, que segue a mesma filosofia de concepção comentada anteriormente. Entre suas funcionalidades pode-se citar capacidades de envio de determinado codec de áudio, envio de vídeo, largura dos quadros de vídeo enviados, definição de salas para conferência, procura por gatekeeper, entre outros. Das bibliotecas utilizadas, o OpenMCU é a mais recente a ser desenvolvida. Por isso, algumas das funcionalidades previstas na Norma H.323 não estão totalmente implementadas, ou mesmo ainda em fase de desenvolvimento. No entanto, isto não impede a utilização desta ferramenta e seu uso cada vez maior pelas pessoas é essencial para que este continue sendo aprimorado.

6.2.3. OpenGatekeeper

A última entidade H.323 utilizada neste projeto é o OpenGate, desenvolvida pelo grupo Egoboo, mas que também necessita das bibliotecas PWLib e OpenH323 para ser compilado e executado. Assim como as bibliotecas anteriores, está disponível em versão para Windows e Linux, e suas características básicas são o controle de registro, admissão e acesso, bem como tradução de endereço e monitoramento de banda [11]. Suas funções avançadas são roteamento de chamadas, suporte a prefixos de gateway, registro e geração de logs de chamadas, armazenamento de informações relativas a gatekeeper vizinhos, além de encerrar a conexão com uma entidade após certo tempo (*time to live).*

Conforme citado anteriormente, o *CVS Repository* dispõe arquivos como os do gong para que sejam baixados pelo usuário. Este repositório possui todas as alterações feitas em qualquer uma

das ferramentas do projeto OpenH323, inclusive as mais recentes que não foram incluídas na última versão das bibliotecas. Assim, para aqueles que estão desenvolvendo algum tipo de trabalho com este recurso, é recomendável consultar periodicamente esta sessão.

6.3. *Ferramenta de Análise de Protocolo* – Sniffer

Sniffers são softwares para captura de pacotes. Sua forma de atuação é bastante variável, inde dezenas de protocolos podem ser analisados, ou apenas um só.

Para que seja possível a transmissão de dados entre duas máquinas numa rede, é necessário que se faça uso dos canais de comunicação, como por exemplo, o cabo que une os micros em uma LAN. Através deste cabo então é que as informações serão trocadas em uma rede local.

Ao longo do cabo de conexão da rede trafegam os dados, divididos em pequenas unidades, denominadas frames (padrão Ethernet). Nestes frames, existem várias seções, onde cada uma contém determinada informação, como os dados de identificação do endereço de origem e destino daquele frame. Com estas informações, o frame busca o adaptador correspondente ao endereço de destino, e faz com que a informação contida seja devidamente entregue. Todos os frames circulam pela rede, ou seja, todos os adaptadores recebem todas as informações que trafegam pela rede, e aquelas que não forem identificadas com os respectivos destinatários são simplesmente descartadas, pois não lhe são devidas.

Um adaptador pode receber e aceitar todos os dados enviados independentemente do destino dos mesmos. Quando a placa de rede funciona desta forma, chamado estado promíscuo, a estação é capaz de capturar todos os pacotes e frames na rede. A ação de capturar os pacotes é denominado *sniffing*.

Neste trabalho, os pacotes visados para esta captura são aqueles definidos pela Recomendação H.323, como o H.225, H.245, entre outros. O sniffer utilizado foi o Ethereal, cujas bibliotecas se encontram disponíveis na Internet. Suas bibliotecas básicas não incluem o padrão H.323, e por este motivo, foi necessário buscar aquelas referentes ao padrão H.323, e compilá-las também.

Esta ferramenta foi bastante útil neste trabalho, principalmente para confirmação e estudo das informações contidas nos frames enviados. Sua tela gráfica apresenta o aspecto da figura 6.1.

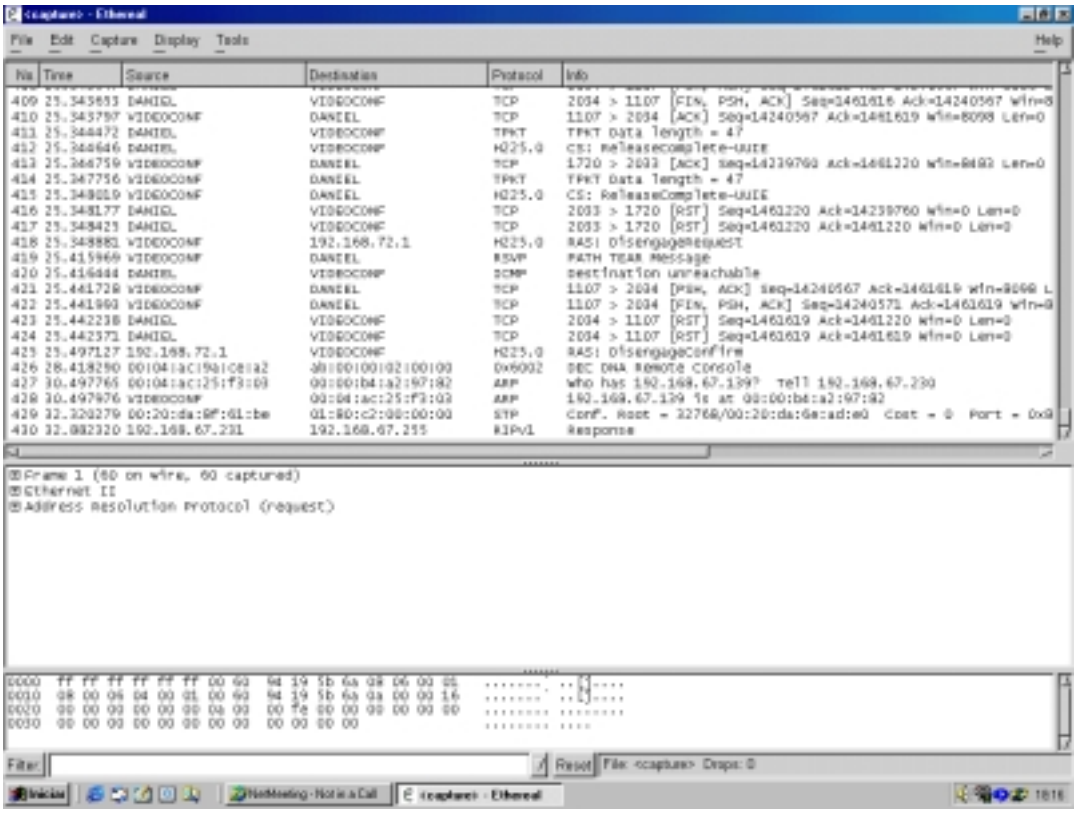

Figura 6.1 – Tela do software Ethereal

6.4. Testes realizados

A opção pela montagem de um ambiente de colaboração visual com ferramentas de software de código aberto está baseada na possibilidade de quem implementa o ambiente, ou beneficiados, de manipular os códigos fonte das entidades. Esta possibilidade é particularmente interessante principalmente quando se visa otimização do sistema.

Um usuário pode, por exemplo, desenvolver um novo processo para utilização do codec de vídeo H.261, como uma nova forma de tratar a imagem baseada na compressão de bits de seu fundo de tela, pois esta é estática. Ou então, implementação de outro codec de vídeo, como o H.263, ainda não disponível nas bibliotecas do OpenH323. Enfim, as possibilidades para manipulação das ferramentas do sistema são enormes e por isso o grande interesse em utilizá-lo.

Outro aspecto interessante a ser observado é o custo que o sistema apresenta, praticamente zero. É claro que são necessários investimentos em microcomputadores para execução das ferramentas, no entanto, o investimento necessário para se montar o ambiente com soluções de hardware seria bem mais elevado.

Devido as bibliotecas do OpenH323 não estarem finalizadas, alguns tipos de chamadas entre entidades não são possíveis até o presente momento, e isto dificulta de certa maneira o perfeito funcionamento do sistema. O fato do processamento de áudio e vídeo ser realizado por software pode também ser impeditivo para determinadas implementações.

Neste trabalho, as entidades H.323 OhPhone, OpenGatekeeper e OpenMCU foram implementadas e testes simples de interoperabilidade foram realizados entre elas. O principal resultado buscado foi o de funcionamento e comunicação, não havendo maior preocupação com boa otimização e modificações nos códigos fonte das bibliotecas.

Nos testes realizados, o sistema operacional utilizado para montagem do ambiente foi o Linux Red Hat 6.2, kernels 2.2.14-5 e 2.2.14-SMP, além das ferramentas *opensource* H.323 foram adquiridas nos sites dos projetos OpenH323 [10] e OpenGatekeeper [11]. As principais ferramentas são pwlib, openh323, opengate, openmcu e ohphone. As bibliotecas pwlib e openh323 são necessárias para compilação de qualquer outra das três outras citadas.

6.4.1. Ambiente 1 - 2 terminias e 1 gatekeeper

Neste teste, fez-se uma chamada entre terminais, com ambos registrados no mesmo gatekeeper, utilizando os equipamentos abaixo:

- 1 computador Pentium 166 MHz, com 128kBytes de memória RAM, software opengatekeeper (gatekeeper - Linux);
- 1 computador Pentium 166 MHz, com 128kBytes de memória RAM, , com uma câmera de vídeo e placa de som, *softwares* OhPhone com front-end gong (terminal 1 - cliente Linux);
- 1 computador Pentium II 300 MHz, com 64kBytes de memória RAM, com uma câmera de vídeo e placa de som, software NetMeeting (terminal 2 - cliente Windows);
- backbone multicast do LabRedes.

A dificuldade inicial neste teste foi entender o funcionamento e o processo de inicialização do OpenGatekeeper. A compilação deste software, apesar de simples, foi difícil de ser realizada, principalmente devido a falta de informação sobre os procedimentos para tanto. A *homepage* do grupo [11], que apresenta este tipo de informação, permaneceu durante vários meses sem possibilidade de ser acessada e por isso, o trabalho não apresentou grande evolução prática neste período. Somente após o retorno do *site* ao funcionamento normal foi possível compilar e executar o OpenGatekeeper.

O ambiente simulado está mostrado na figura 6.2 a seguir.

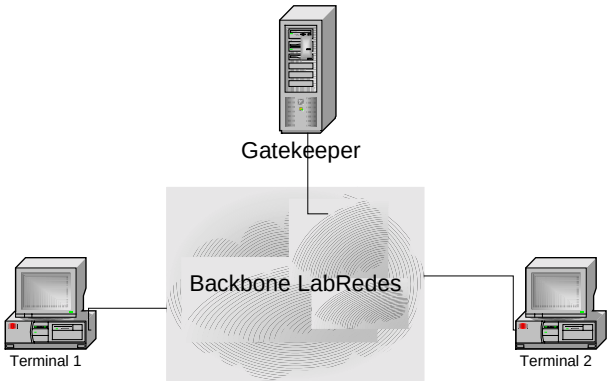

Figura 6.2 – Ambiente com dois terminais e um gatekeeper

O teste consistiu em realizar a chamada do terminal 1 para o terminal 2 e analisar as informações entre eles. Para isto, utilizou-se o *sofware* Ethereal.

Inicialmente são trocadas informações entre o gatekeeper e os terminais para registro dos mesmos. O protocolo utilizado é o H.225, e contém informações de **connectionRequest, connectionConfirm, terminalAlias, rasAddress, terminalType,** entre outros. Constatou-se que as mensagens trocadas seguiam perfeitamente o que estava definido na Recomendação H.323.

Um aspecto interessante é que as chamadas de um terminal a outro, via gatekeeper, pode ser feita pelo endereço IP, nome do terminal ou número dele. Os dois últimos correspondem ao **terminalAlias**.

Conforme comentado anteriormente, o gatekeeper é responsável pelo controle de banda e de acesso. A entidade gatekeeper foi implementada com o opengate, *software* em código aberto, que gera um arquivo para realizar estes controles em /root/.pwlib_config/opengate.ini. Neste estão contidas informações como **Route H245, Local Address, Gatekeeper Id, Endpoint TTL, Neighbours, Log File, Max Bandwidth, Min Bandwidth** e **IsGKRouted.**

Os codecs de áudio trocados entre os terminais seguiram o padrão G.711 e os codecs de vídeo o padrão H.261. A negociação destes codecs foi feita automaticamente entre as entidades envolvidas, sem interferência externa.

Com relação a chamada em si, obteve-se áudio com boa qualidade e atraso bem baixo, inferior a 1s; o vídeo apresentou-se com qualidade média, as imagens eram obtidas com falhas em alguns quadros e atraso maior que o áudio, cerca de 2s. O atraso do vídeo pode ser atribuído principalmente a estar se utilizando uma estação cliente com um processador Pentium de 166 MHz, tendo em vista que é necessário envio e recepção de áudio/vídeo constante, aumentando bastante o processamento da máquina. Ao se realizar o mesmo teste entre duas estações com processador

Pentium II 300 MHz, a comunicação foi realizada em tempo real. A intenção da utilização do processador Pentium 166 MHz foi devido a este possuir a plataforma Linux instalada e assim verificar o funcionamento do terminal implementado em software livre.

6.4.2. Ambiente 2 - 3 terminais e 1 MCU

Neste teste, fez-se a a ligação de 3 terminais ao MCU, de maneira direta. A figura a seguir utilizando os equipamentos:

- 1 computador Compaq Proliant 800, com 512MBytes de memória RAM e dois processadores Pentium Pro 200MHz, *software* OpenMCU (MCU);
- 3 computadores Pentium 166 MHz, com 128kBytes de memória RAM, com uma câmera de vídeo e placa de som, *softwares* OhPhone com front-end gong (terminais 1,2 e 3 - cliente Linux);
- backbone multicast do LabRedes.

O maior impecílio para a realização deste teste foi a escassez de informação acerca do OpenMCU. Devido a esta ser uma biblioteca relativamente recente, informações básicas sobre a entidade são difíceis de serem encontradas, como os procedimentos de compilação e requisitos de hardware. Estas informações foram obtidas após várias correspondências com outros usuários do sistema, os quais também encontraram as mesmas dificuldades. Algumas versões do OpenMCU foram testadas com insucesso, até se chegar a versão utilizada neste projeto.

Outra grande dificuldade encontrada neste teste foi a escolha do processador a ser utilizado. Nas primeiras tentativas, foi utilizado um processador Pentium 166MHz com 128kBytes de memória RAM e as chamadas eram feitas uma a uma. Na primeira chamada o MCU respondia corretamente e estabelecia os canais de comunicação com áudio e vídeo. A segunda chamada também estabelecia os canais de áudio e vídeo mas o processamento do computador já estava bastante alto. Ao se fazer a terceira chamada, o áudio e o vídeo também foram enviados e recebidos corretamente, mas toda a capacidade de processamento da máquina foi utilizada e por isso o sistema cancelou o envio de áudio e vídeo. Devido a este problema, optou-se pela utilização de uma máquina com processador de maior capacidade, além do fato de o equipamento utilizar dois processadores.

O teste consistiu em realizar chamadas dos terminais ao MCU e verificar a divisão da tela do cliente pelo MCU, bem como pacotes trocados e qualidade da sessão estabelecida. A figura 6.3 a seguir ilustra a situação apresentada.

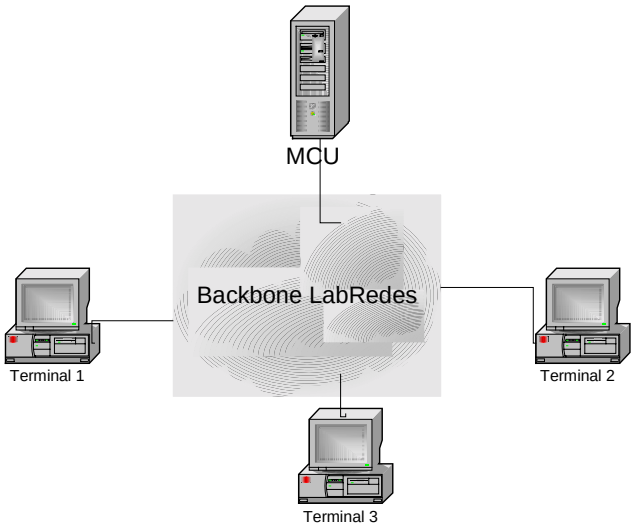

Figura 6.3 – Ambiente com três terminais e um MCU

O registro dos terminais ao MCU, ocorre de maneira semelhante ao gatekeeper, utilizando também o protocolo H.225; informações como **connectionRequest, connectionConfirm, terminalAlias, conferenceAlias, terminalType,** entre outros. Novamente, as mensagens trocadas seguiram o que estava descrito na Recomendação H.323.

A norma H.323 define que a chamada entre terminais e MCU pode ser realizada pelo **conferenceAlias**; no entanto, em testes realizados, não foi possível este tipo de conexão. Isto pode ter ocorrido devido ao código do openmcu não estar com esta implementação. Conforme comentado no item 6.1, esta biblioteca ainda é recente, comparada com as outras do projeto OpenH323 e muitas funcionalidades desta entidade ainda estão sendo desenvolvidas.

O arquivo que controla os parâmetros do MCU também é gerado no diretório /root/.pwlib_config, e é o openmcu.ini. Nele estão contidas informações como **gatekeeper, username, defaultroom, video, videolarge** e **videofill.** As configurações do OpenMCU estão descritas no anexo III e foram configuradas da seguinte maneira:

- defaulroom=labredes;
- username=MCU;
- video=True;
- videotxquality=1;
- videotxfps=20
- videofill=20.

Quando o terminal é registrado ao MCU, são negociados os seguintes parâmetros:

Padrões de canal de áudio enviado e recebido, padrões de vídeo enviado e recebido. Novamente o codec de vídeo foi o H.261 e o codec de áudio G.711. Estes parâmetros também foram negociados automaticamente. A negociação destes está de acordo com as capacidades que cada entidade provê, como codecs G.711, G.729, entre outros, e a escolha dos parâmetros é feita de acordo com os codecs preferenciais e comuns a cada entidade. Este processo pode ser controlado para que se faça a escolha por determinada capacidade.

Nos testes realizados, a qualidade do áudio obtido foi inferior ao da chamada com o gatekeeper, com atraso de cerca de 2s. O vídeo foi mostrado em uma tela dividida em quatro partes, na qual cada terminal ocupava um dos espaços

O aparecimento da imagem se mostrou lento, com atraso da ordem de 3s e imagem não continuada, o que dificulta razoavelmente a interação. No entanto, vale ressaltar que o objetivo do trabalho era a montagem do ambiente, sendo o aspecto qualidade, principalmente utilizando o MCU. Analogamente ao gatekeeper, seus parâmetros podem ser manipulados para melhor aproveitamento do sistema.

Foram realizados ainda, testes neste ambiente com terminais apenas de áudio, sem vídeo, e observou-se que a sessão decorria normalmente. Todos os clientes participavam normalmente da videoconferência, escutando uns aos outros com boa qualidade e atraso baixo, cerca de 1s. Neste caso, o processamento realizado pelo MCU é bem menor àquele com presença também de vídeo.

Realizou-se ainda uma chamada do MCU para um terminal, MCU convida o participante, no entanto algumas mensagens de erro foram exibidas e não foi possível a continuidade da sessão. Isto indica novamente que o código do openmcu não está finalizado e necessita de aprimoramento.

6.5. Futuros testes

Os experimentos realizados neste trabalho consistiram em testes simples para verificação do funcionamento das entidades e posterior montagem do ambiente. Vários cenários de teste não puderam ser implementados devido a falta de conhecimento de procedimentos e falta de tempo hábil para tanto.

Um próximo cenário para ser avaliado é o de 2 terminais, conectados cada um a um gatekeeper diferente, realizando chamada de um para outro, conforme a figura 6.4.

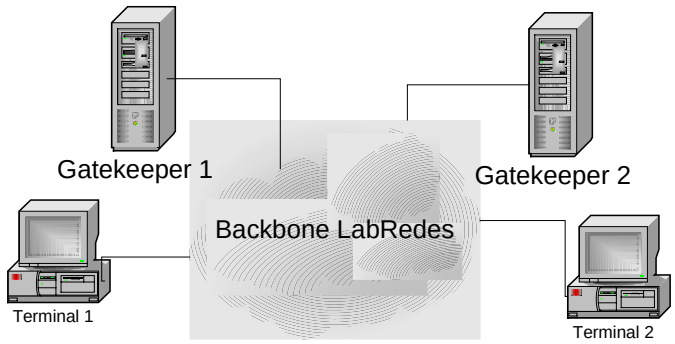

Figura 6.4 – Ambiente com dois terminais e dois gatekeeper

Neste teste, não foram obtidos resultados positivos; cada terminal se registrava em um gatekeeper com sucesso, mas ao se realizar uma chamada de um ponto, seja ela pelo endereço IP, nome do terminal, ou seu número, não era obtido o estabelecimento da sessão. Analisando os pacotes trocados, observou-se que o terminal solicitante fazia o pedido de conexão ao gatekeeper, este fazia um pedido ao outro gatekeeper e então não era obtida resposta. Desta maneira, o gatekeeper inicial retornava uma mensagem de conexão rejeitada.

Outro ambiente possível pode ser composto de um terminal, um gatekeeper e um MCU. Neste caso, não foi possível o estabelecimento da sessão devido ao MCU não ser registrado ao gatekeeper com seu **conferenceAlias**. A Recomendação H.323 define que em sessões envolvendo MCU's e gatekeeper's, as chamadas são estabelecidas por este parâmetro. Sem dúvida, este é um problema no código das bibliotecas e também deve ser aprimorado.

Pode-se também verificar um sistema com três ou mais terminais, dois gatekeepers e um MCU, conforme a figura 6.5.

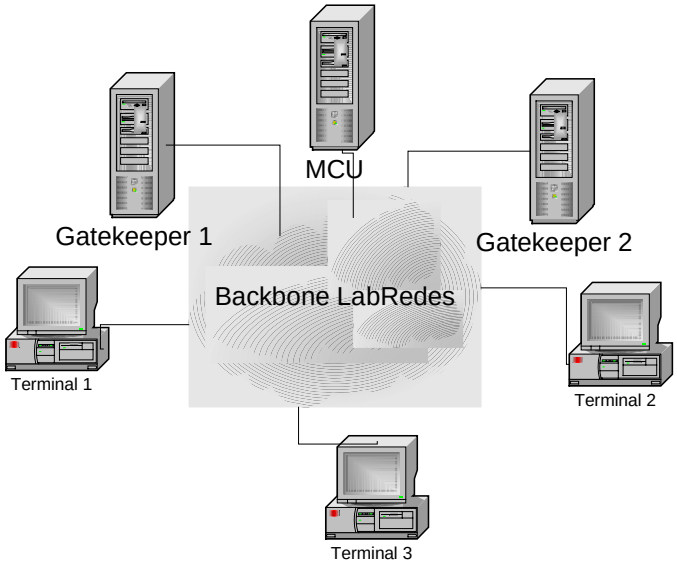

Figura 6.5 – Ambiente com dois terminais e dois gatekeeper e um MCU

Deve ser verificado se não há falha em nenhuma comunicação entre entidades, a boa qualidade de áudio e vídeo, e atraso dos mesmos.

Outra alternativa é a incorporação de um firewall a rede anterior, de maneira a simular dois ambientes de rede distintos, verificando o mascaramento dos pacotes, sua integridade, o estabelecimento das chamadas por todos os terminais e atraso de pacotes. Apesar de não ser uma ferramenta H.323, o firewall deve atuar no sentido de restringir portas de conexão, permitindo apenas o uso das portas *well known* e aquelas utilizadas durante a sessão de videoconferência.

A integração com a rede de telefonia pública comutada (PSTN) deve ser buscada, conforme proposto no capítulo anterior deste trabalho. Enfim, deve-se implementar o sistema de comunicação multiponto proposto neste estudo com todas entidades se comunicando perfeitamente entre si.

Testes com garantia de qualidade de serviço podem também ser realizados no backbone experimental. Neste tipo de serviço ocorre descarte de pacotes de melhor esforço quando há congestionamento na rede. Os dois modelos são utilizados para implementar QoS são os de serviços integrados (IntServ) e de serviços diferenciados (DiffServ). O primeiro está baseado em reserva de recursos, e o segundo é a marcação de pacotes com classes de serviços pré-determinadas.

7. Conclusão

Os experimentos realizados visaram simular e analisar tipos possíveis de sessão de videoconferência. Nos testes realizados, capturou-se os pacotes que eram trocados pelas estações terminais, as portas nas quais cada máquina se comunicava e o tipo de resultado obtido.

Pode-se dizer, no geral, que os ensaios alcançaram êxito em sua plenitude, portanto, atingiram-se os objetivos, que eram o áudio e o vídeo em todas as estações terminais que iriam participar da videoconferência. É válido destacar que a embora a qualidade dos resultados variou de ambiente para ambiente, pode-se manulipular vários parâmetros das entidades envolvidas, buscando otimização do sistema.

Desta maneira, não se pode desprezar a importância deste trabalho no sentido de prover um sistema de videoconferência, na qual vários terminais podem participar de uma única sessão, fato este altamente interessante. Conforme comentado na introdução, as barreiras físicas entre os indivíduos está se estreitando cada vez mais, e a possibilidade de comunicação por meio de videoconferência é, sem dúvida, um dos artifícios que contribuem enormemente para tanto.

É claro que o ambiente simulado não corresponde ao mesmo que será encontrado ao se realizar videoconferência por meio da Internet, onde não se pode garantir principalmente as taxas de transmissão que serão utilizadas, aliado a velocidade de acesso do usuário, pois os pacotes poderão trafegar por rotas distintas e longas, gerando certo atraso e até perda de conteúdo.

Novos testes podem ser realizados com estes equipamentos, implementados com *software* de código aberto ou não, principalmente àqueles que se referem aos realizados com a Internet. Pode-se ainda, verificar se a operação do sistema proposto é eficiente em redes ISDN, como uma rede ATM, e comparar os resultados obtidos com os que obteve-se utilizando um pequeno backbone, como o proposto neste estudo.

Tem-se ainda a possibilidade de novas implementações com modificações feitas no código fonte, alterando propriedades como os codecs de áudio e vídeo, ou até mesmo implementado novos tipos. As possibilidades de trabalhos a serem realizados com *softwares* de código aberto são amplas, e por isso muitas ferramentas ainda aparecerão.

É importante ressaltar que as bibliotecas do projeto OpenH323 não estão finalizadas, ou seja, estão em constante processo de atualização, para aperfeiçoamento e novas implementações no código. Sendo assim, é de grande valia o acompanhamento constante do repositório do projeto para

aquisição de novas versões de arquivos, e até mesmo para o envio para a *homepage* do grupo ao alterações no código feitas pelo usuário, para que possa beneficiar também a outras pessoas, conforme a filosofia do software livre.

Enfim, este projeto pretende contribuir para que novos estudos possam ocorrer na área de videoconferência, uma vez que as entidades H.323 foram implementadas em sua maioria (exceção ao gateway H.323). Sem dúvida, esta será uma área de pesquisa bastante visada por mais alguns anos e, por isso, suas ferramentas básicas devem estar implementadas para que novos estudos possam acontecer.

Anexo I – Pilhas de Protocolos ITU-T Série H

Tabela A.1 – Pilhas de Protocolos ITU-T Série H

Aplicação	**H.320**	**H.321**	**H.322**	**H.323**	**H.324**	**H.310**
Network	Narrowband Switched Digital ISDN	Broadband ISDN ATMLAN	Guaranteed bandwith packet switched networks	Non-guaranteed band width packet switched networks (Ethernet)	PSTN or POTs, the analog phone system	Broadband ISDN ATMLAN
Video	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263	MPEG-2 (H.262) H.261
Audio	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728 G.723 G.729	G.723	MPEG-2 G.711 G.722 G.728
Multiplexing	H.221	H.221	H.221	H.225.0	H.223	H.222.0 H.222.1 (MPEG)
Control	H.230 H.242	H.242	H.242	H.245	H.245	H.245
Multipoint	H.231 H.243	H.231 H.243	H.231 H.243	H.323		
Data	T.120	T.120	T.120	T.120	T.120	T.120

Anexo II – Compilação das Ferramentas do Projeto OpenH323

Compilação das Bibliotecas PWLib e OpenH323

1. Copiar os arquivos pwlib_1.1.32.tar.gz e openh323_1.5.4.tar.gz para o diretório home (não necessariamente precisa ser este);
2. Editar o arquivo profile no diretório /etc e acrescentar as seguintes linhas:

   #PWLIDIR=/home/pwlib

   #export PWLIBDIR

   #OPENH323DIR=/home/openh323

   #export OPENH323DIR

   #LD_LIBRARY_PATH=$PWLIBDIR/lib:$OPENH323DIR/lib

   #export LD_LIBRARY_PATH
3. Reiniciar o sistema para que as linhas de comando sejam executadas;
4. Descomprimir o arquivo pwlib-xxx com o comando:

   #tar –xvzf pwlib_1.1.32.tar.gz
5. Digitar o seguinte comando:

   #cd $PWLIBDIR
6. Executar a compilação das bibliotecas com o comando:

   #make both
7. Após a compilação do pwlib, faz-se a compilação do openh323. Para isto, deve-se digitar:

   #cd /home
8. Descomprimir o arquivo openh323_1.5.4.tar.gz com o comando:

   #tar –xvzf openh323_1.5.4.tar.gz
9. Digitar o seguinte comando:

   #cd $OPENH323DIR
10. Executar a compilação das bibliotecas com o comando:

    # make opt

Compilação do OpenGatekeeper

1. Copiar o opengate.tar.gz para o diretório /openh323;
2. Descomprimir o arquivo com o comando:

   #tar –xzvf opengate.tar.gz
3. Digitar:

   #cd opengate
4. Executar os comandos na seguinte ordem, sempre após concluída a compilação corrente:

   #make optdepend

   #make debugdepend

   #make opt

   #make debug

Compilação do OpenMCU

1. Copiar o openmcu_1.0.7.tar.gz para o diretório /openh323;
2. Descomprimir o arquivo com o comando:

   #tar –xzvf openmcu_1.0.7.tar.gz
3. Digitar:

   #cd openmcu
4. Executar os comandos na seguinte ordem, sempre após concluída a compilação corrente:

   #make optdepend

   #make debugdepend

   #make opt

   #make debug

Compilação do Ohphone e do Front-end gong

1. Copiar o ohphone_1.1.4.tar.gz para o diretório /home;
2. Descomprimir o arquivo com o comando:

   #tar –xzvf ohphone_1.1.4.tar.gz
3. Digitar:

   #cd ohphone
4. Executar os comandos na seguinte ordem, sempre após concluída a compilação corrente:

```
#make optdepend
#make debugdepend
#make opt
#make debug
```

5. O último procedimento é a instalação do *software* gong, um aplicativo gráfico para o ohphone (a instalação prévia do ohphone é necessária). Este *software* encontra-se disponível no repositório das bibliotecas do openh323. Para sua instalação são necessárias algumas bibliotecas do gnome; caso este não esteja instalado no sistema, pode haver problemas de compilação. A compilação do gong segue três passos básicos, constituída pelos seguintes comandos:

```
#./configure
#make
#make install
```

Observação1: O arquivo executável do gong é gerado no diretório /usr/local/bin, e para sua inicialização é necessário que a interface gráfico do Linux esteja sendo utilizada.

Observação2: Antes da execução do ohphone, opengate ou openmcu deve-se ler o *help* (-h) de cada um, para otimização do seu funcionamento. O arquivo executável se encontra sempre no diretório ~/obj_linux_x86_d.

Instalação do Dispositivo de Captura de Vídeo no Linux

Para utilização da câmera de vídeo no Linux, é necessário instalação de seus módulos. O módulo utilizado foi o cpia, versão 1.1, e a câmera modelo webcamII, da creative. O procedimento para instalação da câmera é o seguinte:

1. Descomprimir o arquivo cpia-1.1.tar.gz no diretório home:

```
#tar –xvzf cpia-1.1.tar.gz
```

2. Compilar o módulo com o comando:

```
#make
```

3. Serão gerados os objetos cpia.o e cpia_pp.o no diretório /module. Dentro deste diretório, executa-se o comando a seguir para inicialização dos módulos:

```
#./loadpp
```

Anexo III – Sintaxe de Comando para as Ferramentas do OpenH323

Parâmetros utilizados para o OpenMCU

-u --username str : Set the local endpoint name to str

-g --gatekeeper host: Specify gatekeeper host.

-n --no-gatekeeper : Disable gatekeeper discovery.

--require-gatekeeper: Exit if gatekeeper discovery fails.

-i --interface ip : Bind to a specific interface

--g711frames count : Set the number G.711 frames in capabilities (default 30)

--gsmframes count : Set the number GSM frames in capabilities (default 4)

-t --trace : Enable trace, use multiple times for more detail

-o --output : File for trace output, default is stderr

--save : Save arguments in configuration file

-v --video : Enable H261 video handling

--videolarge : Set the video size from normal (176x144) to large (352x288). --videotxquality n : Select sent video quality,(def 9). 1(good)<=n<=31

--videofill n : Select number of updated background blocks per frame 1<=n<=99 (2 def)

--videotxfps n : Maximum number of transmitted video frames per sec 1<10(def)<30

--defaultroom name : Connections without a room name will join this room

(Default room is room101)

--no-defaultroom : Reject connections with no room specified

--disable-menu : Disable the command line menu

-h --help : Display this help message

Parâmetros utilizados para o OhPhone

-a --auto-answer : Automatically answer incoming calls

-d --autodial host : Autodial host if phone off hook

-h --help : Display this help message.

-l --listen : Only listen for incoming calls

-v --verbose n : Set amount of information displayed (0=none)

--disable-menu : Disable internal menu

--ringfile filename : Set sound file for "ring" annunciation

--ringdelay seconds : Set delay between playing above file

--save : Save parameters in configuration file.

Gatekeeper options:

-g --gatekeeper host : Specify gatekeeper host.

-G --gatekeeper-id name : Specify gatekeeper by ID.

-n --no-gatekeeper : Disable gatekeeper discovery.

-r --require-gatekeeper : Exit if gatekeeper discovery fails.

-p --proxy host : Proxy/Gateway hostname/ip address

Divert options:

-F --forward-always party : Forward to remote party.

-B --forward-busy party : Forward to remote party if busy.

-N --forward-no-answer party : Forward to remote party if no answer.

--answer-timeout time : Time in seconds till forward on no answer.

Protocol options:

-i --interface ipaddr : Select interface to bind to for incoming connections (default is all interfaces)

--listenport : Port to listen on for incoming connections (default 1720)

--connectport port : Port to connect to for outgoing connections (default 1720)

--connectring num : Distinctive ring number to send to remote - 0 (default) to 7

-b --bandwidth bps : Limit bandwidth usage to bps bits/second

-f --fast-disable : Disable fast start

-T --h245tunneldisable : Disable H245 tunnelling.

-u --user name : Set local alias name(s) (defaults to login name)

--tos n : Set IP Type of Service byte to n

--setup-param string : Arbitrary data to be put into H.225 Setup PDU

Audio options:

-e --silence : Disable silence detection for GSM and software G.711

-j --jitter delay : Set jitter buffer to delay milliseconds

--recvol n : Set record volume

--playvol n : Set play volume

Video transmit options:

--videodevice dev : Select video capture device (default /dev/video0)

--videotransmit : Enable video transmission

--videolocal : Enable local video window

--videosize size : Sets size of transmitted video window size can be small (default) or large

--videoformat type : Set capture video format can be auto (default) pal or ntsc

--videoinput num : Select capture video input (default is 0)

--videotxquality n : Select sent video quality,(def 9). 1(good)<=n<=31

--videofill n : Select number of updated background blocks per frame 2(def)<=n<=99

--videotxfps n : Maximum number of transmitted video frames per sec 2<10(def)<30

Video receive options:

--videoquality n : Set received video quality hint - 0 <= n <= 31

--videoreceive viddev : Receive video to following device

: null do nothing

: ppm create sequence of PPM files

: svga256 256 colour VGA (Linux only)

: svga full colour VGA (Linux only)

: x11 automatically pick best X11 mode

: x1124 X11 using 24 bit colour

: x1116 X11 using 16 bit colour

: x118 X11 using 8 bit grey scale

--videopip : Local video is displayed in corner of received video

Sound card options:

-s --sound device : Select sound card input/output device

--sound-in device : Select sound card input device (overrides --sound)

--sound-out device : Select sound card output device (overrides --sound)

--sound-buffers n : Set sound buffer depth (default=2)

--sound-mixer device : Select sound mixer device (default is /dev/mixer)

--sound-recchan device : Select sound mixer channel (default is mic)

--sound-recvol n : Set record volume for sound card only (overrides --recvol)

--sound-playvol n : Set play volume for sound card only (overrides --playvol)

Quicknet card options:

-q -quicknet dev : Use device (number or full device name)

-C --country name : Set the country code for Quicknet device

--aec n : Set Audio Echo Cancellation level (0..3)

--autohook : Don't use hook switch (for PhoneCard)

-c --callerid : Enable caller id display

--calleridcw : Enable caller id on call waiting display

--dial-after-hangup : Present dial tone after remote hang up

--quicknet-recvol n : Set record volume for Quicknet card only (overrides recvol)

--quicknet-playvol n : Set play volume for Quicknet card only (overrides playvol)

Audio Codec options:

-D --disable codec : Disable the specified codec (may be used multiple times)

-P --prefer codec : Prefer the specified codec (may be used multiple times)

--g711frames count : Set the number G.711 frames in capabilities (default 30)

--gsmframes count : Set the number GSM frames in capabilities (default 4)

--g7231 : Set G.723.1 as preferred codec

--gsm : Set GSM 06.10 as preferred codec (default)

--g711-ulaw : Set G.711 uLaw as preferred codec

--g711-alaw : Set G.711 ALaw as preferred codec

--g728 : Set G.728 as preferred codec

--g7231 : Set G.723.1 as preferred codec

Debug options:

-t --trace : Enable trace, use multiple times for more detail

-o --output : File for trace output, default is stderr

--setallocationbreakpoint n : Enable breakpoint on memory allocation n

Parâmetros utilizados para o OpenGatekeeper

-h --help output this help message and exit

-v --version display version information and exit

-d --daemon run as a daemon

-u --uid uid set user id to run as

-g --gid gid set group id to run as

-p --pid-file name or directory for pid file

-t --terminate orderly terminate process in pid file

-k --kill preemptively kill process in pid file

-c --console output messages to stdout rather than syslog

-l --log-file file output messages to file or directory instead of syslog

-x --execute execute as a normal program

-i --ini-file set the ini file to use, may be explicit file or a ':' separated set of directories to search.

-C --core-size set the maximum core file size

Referências Bibliográficas e Eletrônicas

[1] Sauer, C.; Duran J.: "*Mainstream Videoconferencing : A Developer's Guide to Distance Multimedia*", Ed.Paperback, 1997.

[2] Wilcox, James R.:Videoconferencing: "*The Whole Picture*", Ed. Paperback, 2000.

[3] Conference, North American Serials Interest Group, "*From Carnegie to Internet2: Forging the Serial's Future"*, Ed. Paperback, 1999.

[4] Jr., José Furst, "Telefonia IP, O Novo Mundo das Telecomunicações", Cisco Systems, Inc. 1999.

[5] Perey, Chistine, "*H.323 Videoconferencing Standard*", Chapman & Hal,l 1998.

[6] Dutta-Roy, Amitava, "*Virtual Meetings with Desktop Conferencing*", IEEE Spectrum, 1998.

[7] http://www.vovida.com

[8] http://www.mozilla.org/NPL/MPL-1.0.html

[9] http://www.cybertechmedia.com/compare.html, CyberTech Media Group, "*Streaming Video Production and Bandwidth Requirements*", 2000.

[10] http://www.openh323.org, "*OpenH323 Project"*, Equivalence Pty Ltd., 1998.

[11] http://www.opengatekeeper.org, "*OpenGatekeeper"*, Egoboo, 2000.

[12] http://www.opensource.org, Hunka, George, "*OpenSource"*, 2001.

[13] http://www.packetizer.com, "*Packetizer*", Packetizer Inc, 2000.

[14] http://www.dtic.mil/ieb_cctwg/contrib-docs/VTC001/toc.htm, Corporation for Open Systems International, 1994.

[15] http://www.ethereal.com, "*Sniffing the glue that holds the Internet together"*, 2000.

[16]http://whipper.uwc.ac.za/staff/btucker/courses/csn73a/Reports/H323/index.htm

[17] http://www.h323.org, "*Free H.323"*, 2001.

[18] http://www.protocols.com/pbook/h323.htm , "*H.323*", Radcom Inc., 1998.

[19] http://www.cisco.com/networkers/nw00/pres/2005.pdf

[20] http://www.databeam.com/h323/h323primer.html, Databeam, "*A Primer on the H.323 Series Standard"*, 1997.

[21] http://www.cis.ohio-state.edu/~jain/cis788-99/h323/index.html, Asim Karim, 2000.

[22] http://www.linuxdoc.org/HOWTO/Firewall-HOWTO.html, Mark Grennan, 2000.

[23] http://www.linuxdoc.org/HOWTO/IP-Masquerade-HOWTO.html, David Ranch, 2000.

[24] http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html, Rusty Russell, 2000.

[25] http://www.imtc.org/standards.htm, *"Standards"*, 2001.

[26] http://developer.intel.com/technology/itj/q21998/pdf/h323.pdf, Toga, James & ElGebaly, Hani, *"Demystifying Multimedia Conferencing Over the Internet Using the H.323"*, 1998.

[27] http://fwup.org, *"Firewall"*, 2000.

[28] http://sourceforge.net. *"Installer Une WebCamII (creative)"*, 1999.

[29] http://webcam.sourceforge.net.

[30] http://www.microsoft.com/windows/netmeeting, Microsoft.

[31] http://www.equival.com, Equivalence Pty Ltd, 2001.

[32] www.elektroindonesia.com/elektro/tab9a1.html, Electronic Design, 1996.

[33] Recomendação H.225 - *Call Signalling Protocols and Media Stream Packetization for Packet-Based Multimedia Communication Systems*, 07/2000.

[34] Recomendação H.245 - *Control Protocol for Multimedia Communication*, 06/2000.

[35] Recomendação H.323 - *Packet-Based Multimedia Communications Systems*, 07/2000.

[36] Recomendação Q.931 - *ISDN User-Network Interface Layer 3 Specification for Basic Call Control*, 08/1998.

Glossário

Canal lógico H.245: Canal que leva fluxos de informação entre dois terminais H.323. É usado um canal confiável para H.245.

Canal RAS: Canal não confiável que carrega registro, admissão e mensagem de estado da largura de banda disponível entre duas entidades H.323.

Conferência Multiponto: Conferência entre três ou mais terminais que podem estar numa LAN ou rede de comutação de circuitos.

E.164: formato de endereços para redes RDSI. Consulte Recomendação ITU E.164 (1991).

Controlador Multiponto (MC): Entidade que provê controle de três ou mais terminais numa conferência multiponto.

Endereço de transporte: A combinação de um endereço de rede e um que identifica um terminal de nível de transporte, por exemplo um endereço de IP.

Entidade H.323: Qualquer componente H.323, inclusive terminais, gateways, gatekeeper, MCs, MPs, e MCUs.

Gatekeeper (GK): Uma entidade H.323 que provê tradução de endereço, controle acesso, e administração de largura de banda na LAN para terminais H.323, gateways, e MCUs.

Gateway (GW): Uma entidade H.323 que provê em tempo real, interoperabilidade entre terminais H.323 na LAN e outros terminais ITU

H.323: Norma adotada para comunicar em qualquer rede de comutação de pacotes, inclusive a Internet.

Multicast: Processo de transmitir de uma fonte a muitos destinos

Pacote RTP: Pacote de dados que consiste no header de RTP fixo, uma lista possivelmente vazia de fontes contribuintes e os seus dados.

Pacote de RTCP: Pacote de controle que consiste em uma parte de cabeçalho fixa semelhante ao de RTP , seguido por elementos estruturados que variam e dependem do tipo de pacote RTCP.

Processador Multiponto (MP): Entidade que provê processamento de áudio e vídeo na conferência multiponto. O MP implementa mistura e comutação, ou outro processo no fluxos de bits.

Protocolo de Reserva de recurso (RSVP): Especificação do IETF. Permite aplicações como largura de banda dedicada.

Q.931: Protocolo de sinalização de chamada para setup e terminação de chamadas.

Qualidade de Serviço (QoS): Garantia de disponibilidade de largura de banda na rede, para aplicações.

RTP: Protocolo de Tempo Real. Utilizado sobre o IP em aplicações com características de tempo real.

RTCP: Protocolo de Controle de Tempo Real: Permite aplicações para sincronizar e deteriorar informação áudio e vídeo, bem como outras funções de controle.

Rede digital de serviços integrados (RDSI)

Rede projetada para melhorar os serviços de telecomunicações do mundo providenciando um padrão internacionalmente aceito para transmissão de voz, dados e sinalização; propõe todos os circuitos de transmissão fim a fim digitais.

Terminal: é um ponto terminal que providencia comunicação em tempo real com outro terminal, gateway, ou MCU.

Transmissão confiável: Transmissão de dados orientada à conexão, garantindo sequência, informação sem erros, e transmissão que controla o fluxo de mensagens para o receptor.

Transmissão não confiável: transmissão que providencia entrega de pacotes utilizando "melhor esforço". Podem ser perdidas mensagens transmitidas pelo remetente, ou podem ser recebidas fora de sequência.

União de Telecomunicações internacional (ITU): Organização estabelecida pelas Nações Unidas para fixar padrões de telecomunicações, e alocar frequências para vários usos.

Unidade controle multiponto (MCU): Permite três ou mais terminais e gateways participarem de uma conferência multiponto. O MCU inclui controlador multiponto obrigatório e processadores multiponto opcionais.